

# Aufbau der Rule Struktur

Dieses Dokument beschreibt den Aufbau der XML Rule Struktur des ELOas. Diese Struktur wird im Normalfall über ein grafisches User Interface gepflegt werden. Falls trotzdem manuell eingegriffen werden muss, kann diese Beschreibung als Referenz verwendet werden. Ebenso dient diese Beschreibung als Referenz für die Implementierung des GUI.

## Inhalt

1	Allgemeiner Aufbau .....	2
1.1	Alle Einträge im <base> Abschnitt .....	3
1.2	Alle Einträge im <search> Abschnitt .....	3
1.3	Alle Einträge im <rule> Abschnitt .....	4
2	Anmerkungen .....	7
3	Beispielsstruktur .....	9

## 1 Allgemeiner Aufbau

Die komplette Struktur ist in ein Tag <ruleset> eingebettet. Dieses besteht aus zwei Teilen, einem <base> Eintrag am Anfang, gefolgt von beliebig vielen <rule> Einträgen.

Der <base> Eintrag beinhaltet die Informationen zur Suche der zu bearbeitenden Einträge. Diese umfassen die Suchezeile, den Suchbegriff, Masken und Datumseinschränkungen.

Die <rule> Einträge beinhalten jeweils eine Verarbeitungsvorschrift. Jede Regel kann mit einer Bedingung belegt werden, sie kann das Ablageziel verändern oder Inhalte der Indexzeilen anpassen. Zudem kann eine Rule auch einen JavaScript Inhalt haben. Wenn dieser definiert ist, dann werden die anderen Einträge ignoriert, sie dürfen aber mit Werten gefüllt bleiben.

Wenn die Bedingung einer Regel „OnError“ lautet, dann handelt es sich bei dieser Regel um eine Fehlerbehandlungs-Regel. Nach jeder Regel darf eine Fehlerbehandlung erfolgen, ganz am Ende muss eine Fehlerbehandlungs-Regel eingetragen werden. Diese abschließende Fehlerregel wird auch dann aufgerufen, wenn beim Verschieben oder Speichern ein Fehler auftritt. Wenn bei der Bearbeitung innerhalb einer normalen Regel ein Fehler auftritt, wird die nächstmögliche Fehlerbehandlungs-Regel aufgerufen und danach die Bearbeitung abgebrochen.

Beispiel für einen einfachen Ruleset:

```
<ruleset>
  <base>
    <name>Name des Rulesets</name>
    <search>
      <name>Name der Indexzeile im JavaScript Code</name>
      <value>Suchbegriff im JavaScript Code</value>
      <mask>Nummer der Dokumentenmaske für die Suche</mask>
    </search>
    <interval>5M</interval>
  </base>
  <rule>
    <name>Name der Regel</name>
    <destination mask="Ordnermaske"> Neues Ziel im JavaScript Code</destination>
    <index>
      <name>Name der Indexzeile im JavaScript Code</name>
      <value>Neuer Inhalt der Indexzeile im JavaScript Code</value>
    </index>
  </rule>
  <rule>
    <name>Name der Fehlerbehandlungs-Regel</name>
```

```
<condition>OnError</condition>
</rule>
</ruleset>
```

## 1.1 Alle Einträge im <base> Abschnitt

Tag	Funktion	Beispiel
name	Name des Rulesets. Dieser Name wird auf der Statusseite angezeigt, er wird aber nicht weiter verarbeitet.	SAP Verarbeitung
search	Parameter für die Suche der zu verarbeitenden Dokumente. Beschreibung siehe „Alle Einträge im <search> Abschnitt“.	
masks	Wenn im Rahmen der Verarbeitung auf einen anderen Ablagemaskentyp umgeschaltet werden soll, dann muss hier eine Liste aller möglichen Zielmaskennummern aufgeführt werden. Jedes Maskennummer wird durch ein <mask> Tag umrahmt.	<mask>3</mask> <mask>4</mask>
interval	Wiederholungsintervall für die Abarbeitung der Suche. Dieses Intervall kann in Minuten (5M) oder Stunden (1H) angegeben werden. Weiterhin kann es einmal pro Tag zu einer bestimmten Uhrzeit (15:30) ausgeführt werden, einmal pro Wochentag (17:20/SA) oder einmal pro Monat (22:00/31).  Falls bei der monatlichen Ausführung ein Tag angegeben wird, der im aktuellen Monat nicht vorkommt (z.B. der 31. In einem Februar), dann wird jeweils der letzte Tag des Monats verwendet.	5M 1H 15:30 17:20/SA 22:00/31

## 1.2 Alle Einträge im <search> Abschnitt

Die Einträge im Abschnitt <search> bestimmen welche Dokumente bearbeitet werden. Zum Beginn jedes Durchlaufs wird eine Suche mit diesen Parametern ausgeführt. Die Trefferliste wird dann anhand der Regeln bearbeitet.

```
<search>
  <name>Name der Indexzeile im JavaScript Code</name>
  <value>Suchbegriff im JavaScript Code</value>
  <mask>Nummer der Dokumentenmaske für die Suche</mask>
```

<max>Maximale Anzahl der Dokumente pro Durchlauf</max>  
</search>

Tag	Funktion	Beispiel
name	Name der Indexzeile im JavaScript Code. Wenn der Name feststehend ist, kann direkt ein Text in Anführungszeichen eingegeben werden. Es kann aber auch ein beliebiger JavaScript Ausdruck verwendet werden.	"ELOOUTL2"
value	Suchbegriff im JavaScript Code. Wenn der Wert feststehend ist, kann direkt ein Text in Ausführungszeichen eingegeben werden. Es kann aber auch ein beliebiger JavaScript Ausdruck verwendet werden.	"ELO*"
mask	Nummer der Dokumentenmaske für die Suche. Es kann hier nur eine Dokumentenmaske, keine reine Suchmaske verwendet werden, da beim Einlesen davon ausgegangen wird, dass alle Treffer die gleiche Maskendefinition besitzen.	2
max	Maximale Anzahl der Dokumente pro Durchlauf bei der Suchabfrage zum Indexserver. Falls mehr Treffer vorliegen, werden diese in einem weiteren Durchlauf bearbeitet nachdem alle anderen Rulesets durchlaufen wurden. Damit soll verhindert werden, dass ein extrem umfangreicher Ruleset die Abarbeitung aller anderen Ruleset unterdrückt.  Es sind maximal 1000 Dokumente pro Durchlauf zulässig.	200
idate xdate	Die Trefferliste kann durch ein Datumsbereich im Ablage- (idate) oder Dokumentendatum (xdate) eingeschränkt werden. Dieses Datum kann entweder in absoluten Werten im ISO Datumformat (YYYYMMTT) oder in relativen Werten zum aktuellen Tag (-5) eingetragen werden.  Der Bereich besteht jeweils aus einem Startdatum in einem <from> Tag und einem Endedatum in einem <to> Tag.	<idate> <from>-5</from> <to>+0</to> </idate>

### 1.3 Alle Einträge im <rule> Abschnitt

Nach dem <base> Abschnitt kommen beliebige viele <rule> Abschnitte. Diese werden bei der Abarbeitung in der Reihenfolge der Definition durchlaufen.

Eine <rule> kann in zwei unterschiedlichen Ausprägungen vorliegen: als normale Regel und als Fehlerregel. So eine Fehlerregel wird in der normalen Ausführung einfach übersprungen. Nur im Fehlerfall wird die nächste verfügbare Fehlerregel in der Reihenfolge aufgerufen, danach wird die Abarbeitung für dieses Dokument abgebrochen, d.h. – nach einer Fehlerregel werden keine weiteren Regeln abgearbeitet.

Die letzte Regel in der <rule> Kette muss immer eine Fehlerregel sein. Damit ist sicher gestellt, dass auf jeden Fall eine Fehlerbehandlung zur Verfügung steht. Zudem wird diese Regel aufgerufen, wenn beim Verschieben oder Speichern ein Fehler auftritt.

Tag	Funktion	Beispiel
name	Name der Regel, wird nur zur besseren Lesbarkeit und zur Dokumentation benötigt.	Indexergänzung
condition	Ausführungsbedingung für diese Regel. Falls es sich um eine Fehlerregel handelt, wird hier der feste Text „OnError“ eingetragen. Hier ist auf eine korrekte Schreibweise zu achten, andernfalls wird die Regel nicht als Fehlerregel erkannt.  Die Ausführungsbedingung liegt als JavaScript Code vor. Nur, wenn die Bedingung „true“ ist, wird die Regel ausgeführt.	KDNR == "123"
destination	Neues Ablageziel des Dokuments als Archivpfad. Der Eintrag ist optional und kann leer bleiben, in diesem Fall bleibt das Dokument an seiner ursprünglichen Position. Falls es mehrere Regeln mit destination gibt, wird das erste Ziel als neuer Ablageort verwendet, alle weiteren Ziele werden zusätzlich als Referenzen eingetragen.  Wenn ein Ablageziel noch nicht vorhanden ist, wird es automatisch angelegt.  Das destination Tag kann ein zusätzliches Attribut „mask“ enthalten, welches die Nummer der Ordnermaske für neu erzeugte Ordner enthält. Wenn dieses Attribut nicht vorhanden ist, wird als default die „1“ verwendet, die Nummer der Ordnermaske in einem Standardarchiv.	<destination mask="1"> ¶ELO¶Mails¶ + ELOOUTL1</destination>
mask	Neue Verschlagwortungsmaske des Dokuments. Wenn dieser Eintrag ist vorhanden ist oder die	<mask>20</mask>

	<p>Maskennummer -1 ist, bleibt die ursprüngliche Verschlagwortungsmaske erhalten.</p> <p>Wenn die Maske gewechselt wird, werden automatisch alle Einträge mit gleichem Gruppennamen übernommen. Das wird auch dann korrekt ausgeführt, wenn die Indexzeilenaufteilung unterschiedlich ist.</p> <p>Falls die ursprüngliche Ablagemaske Indexzeilen enthielt, die die neue Maske nicht besitzt, werden diese Daten automatisch und ohne Fehlermeldung verworfen.</p> <p>Der ELOas kann keine Dokumente bearbeiten, deren Ablagemaske den gleichen Gruppennamen für mehrere Einträge verwendet, da die interne Verarbeitung und die Struktur der Regeln von einer eindeutigen Zuordnung ausgehen.</p>	
index	<p>Innerhalb einer Regel kann es beliebig viele Indexeinträge geben. Jeder Indexeintrag enthält den Namen der betroffenen Indexzeile und einen JavaScript Ausdruck mit dem neuen Wert.</p> <p>Bei Indexzeilen mit ISO-Datum und den Feldern für das Ablage- und Dokumentendatum muss die Eingabe auch im ISO Datumsformat erfolgen.</p> <p>Neben den Indexfeldern mit den Gruppennamen der Suchmaske stehen alle Gruppennamen der alternativen Ablagemasken und eine Reihe von Pseudo-Indexzeilen mit einigen Standardwerten der Verschlagwortung zur Verfügung:</p> <p>NAME: Kurzbezeichnung</p> <p>DOCDATE: Dokumentendatum</p> <p>ABLDATE: Ablagedatum</p> <p>ARCHIVINGMODE: Revisionsstatus 0, 1 oder 2 für Freie Bearbeitung, Versionskontrolliert oder Revisionssicher.</p> <p>ACL: mit „PARENT“ die ACL des neuen Ablageziels übernehmen. Mit &lt;Rechte&gt;:&lt;Name&gt; beliebige</p>	<pre>&lt;index&gt; &lt;name&gt;DOCDATE&lt;/name&gt; &lt;value&gt;"20070930"&lt;/value&gt; &lt;/index&gt;</pre>

	Gruppenberechtigungen definieren.  OBJCOLOR: Farbnummer des Eintrags  OBJDESC: Zusatztext  OBJTYPE: Dokumenten- oder Ordnerartyp des Eintrags. Achtung: durch eine falsche Zuordnung kann es zu Störungen bei der weiteren Verarbeitung kommen. Dokumente dürfen nur einen OBJTYPE zwischen 254 und 286 besitzen.	
script	Eine Regel kann auch direkt den auszuführenden JavaScript Code enthalten. In diesem Fall werden allen anderen Parameter dieser Regel ignoriert, sie dürfen aber (z.B. zu Dokumentationszwecken) weiter enthalten sein.	

## 1.4 Berechtigungsänderungen

In der Pseudo-Indexzeile ACL kann eine geänderte Berechtigung hinterlegt werden. Im einfachsten Fall trägt man hier „PARENT“ ein, dann wird beim Speichern die Berechtigung des Zielordners für diesen Eintrag übernommen. Es kann aber auch eine komplette Berechtigungsliste hinterlegt werden. Diese Liste besteht aus einer Folge von Einzelberechtigungen die durch ein Pilcrow Symbol getrennt werden. Jede Einzelberechtigung besteht aus der Berechtigungsmaske (RWDEL – Read, Write, Delete, Edit, List) gefolgt von einem Doppelpunkt und dem Gruppennamen. Bei UND Gruppen wird statt des einfachen Gruppennamen eine Folge von Namen, getrennt jeweils auch durch einen Doppelpunkt angegeben.

R:Jeder¶RW:Controlling¶RWDEL:Verwaltung:Stuttgart:Leitungsteam

Im Beispiel erhält die Gruppe „Jeder“ einen Lesezugriff, die Gruppe „Controlling“ Lese- und Schreibzugriff und die UND-Gruppe „Verwaltung und Stuttgart und Leitungsteam“ Vollzugriff auf das Dokument.

Falls eine Berechtigung auf einen Anwender statt auf eine Gruppe gesetzt werden soll, muss in die Berechtigungsliste zusätzlich noch ein „U“ aufgenommen werden.

UR:Administrator

## 2 Anmerkungen

Bei der Generierung des JavaScript Codes werden alle Gruppennamen der Suchmaske und der alternativen Ablagemasken als Variable in vollständiger Großschreibweise angelegt. Dieses

Verfahren minimiert das Risiko, dass sich Gruppennamen mit Standardbezeichnern aus JavaScript oder der ELO Laufzeitumgebung überschneiden. Prinzipiell kann es aber trotzdem zu Problemen kommen, falls einer der Gruppennamen identisch zu einem Standardbezeichner oder einer der Übersetzungslisten ist.

```
var NAME;  
var ARCDATE;  
var DOCDATE;  
var OBJCOLOR;  
var OBJDESC;  
var OBJTYPE;  
var ARCHIVINGMODE;  
var ACL;  
var EM_PARENT_ID;  
var EM_PARENT_ACL;  
var EM_SEARCHNAME;  
var EM_SEARCHVALUE;  
var EM_SEARCHCOUNT;  
var EM_SEARCHMASK;  
var EM_IDATEFROM;  
var EM_IDATETO;  
var EM_XDATEFROM;  
var EM_XDATETO;  
var EM_FOLDERMASK = "1";
```

Hinweis: diese Liste kann sich im Laufe des Projektfortschritts erweitern, insbesondere kann sie auch durch lokales Customizing um zusätzliche Einträge ergänzt werden.

Die Nummer der Ablagemaske des aktuellen Dokuments kann über eine Regel geändert werden. Falls es sich dabei um eine ungültige Maskennummer oder um eine Nummer handelt, die nicht in der Liste der alternativen Zielmasken aufgeführt wurde, erzeugt das erst beim Speichern des Dokuments einen Laufzeitfehler und nicht direkt bei der Maskenzuweisung.

Wenn aufgrund eines Laufzeitfehlers eine Fehlerregel aufgerufen wird, dann löscht diese alle bereits vorbelegte Ablageziele der zuvor abgearbeiteten Regeln. Wenn die Fehlerregel keine eigene <destination> besitzt, bleibt das Dokument an seiner ursprünglichen Position, andernfalls wird das Ziel der Fehlerregel verwendet.

Das Verschieben und die Speicherung der geänderten Verschlagwortung passieren erst am Ende nach der Abarbeitung der letzten Regel. Wenn es dabei zu einem Fehler kommt, wird also die letzte Fehlerregel aufgerufen und nicht die Fehlerregel, die zu der Regel gehört, die das Ziel festgelegt hat (was aber identisch ist, wenn es nur eine einzige Fehlerregel gibt).



### 3 Beispielsstruktur

Im Folgenden wird eine beispielhafte Definition aufgeführt, dazu wird der generierte Code aufgelistet. Diese Information dient nur der Orientierung.

```
<ruleset>
  <base>
    <name>Mailmaske Thiele</name>
    <search>
      <name>"ELOOUTL2"</name>
      <value>"Thiele*"</value>
      <mask>2</mask>
      <max>2</max>
      <idate>
        <from>"-35"</from>
        <to>"1"</to>
      </idate>
    </search>
    <masks>
      <mask>12</mask>
      <mask>13</mask>
      <mask>20</mask>
    </masks>
    <interval>1M</interval>
  </base>
  <rule>
    <name>Regel 1</name>
    <destination mask="5">"¶Thiele¶Mails¶" + ELOOUTL1</destination>
    <mask>20</mask>
    <index>
      <name>ADDENTRY</name>
      <value>getObjShort(2)</value>
    </index>
    <index>
      <name>ELOOUTL2</name>
      <value>"!!" + ELOOUTL2</value>
    </index>
    <index>
      <name>DOCDATE</name>
      <value>"20070930"</value>
    </index>
    <index>
      <name>ARCHIVINGMODE</name>
      <value>2</value>
    </index>
  </rule>
</ruleset>
```

```
</index>
<index>
  <name>ACL</name>
  <value>"PARENT"</value>
</index>
</rule>
<rule>
  <name>Journal-Kopie</name>
  <destination mask="1">"¶Thiele¶Journale¶" + ELOOUTL1</destination>
</rule>
<rule>
  <name>Script rule</name>
  <script>
    moveTo(Sord, "¶Ablage¶Ziele1¶" + ELOOUTL1);
    moveTo(Sord, "¶Ablage¶Ziele2¶" + ELOOUTL2);
    moveTo(Sord, "¶Ablage¶Ziele3¶" + ELOOUTL3);
  </script>
</rule>
<rule>
  <name>Global Error Rule</name>
  <condition>OnError</condition>
  <destination>"¶Thiele¶Error"</destination>
  <index>
    <name>ELOOUTL2</name>
    <value>"!!" + ELOOUTL2</value>
  </index>
  <index>
    <name>ARCHIVINGMODE</name>
    <value>0</value>
  </index>
</rule>
</ruleset>
```