

ELOas-Debugger (Java FX)

[Stand: 21.04.2016 | Programmversion: 10.00.000]

Das folgende Dokument zeigt die wichtigsten Funktionen des ELOas Debugger, der mit Hilfe von Java FX implementiert wurde. Der ELOas Debugger wird über die Datei *EloAsDebug.exe* gestartet und setzt voraus, dass mindestens JRE 1.7 auf dem Rechner installiert ist.

Inhalt

1	Oberfläche	2
1.1	Statusmeldungen.....	3
1.2	Programminformationen	3
2	ELOas Debugger starten.....	4
3	Profile	5
3.1	ELOas Debugger-Profil verwenden.....	5
3.2	Profilübersicht	6
3.3	Informationen zum gewählten Profil	7
3.4	Profile bearbeiten.....	8
3.5	Bearbeiten von direkten Regeln	9
4	Tastaturkombinationen	10
5	Java-Bibliotheken	11

1 Oberfläche

Das Hauptfenster vom ELOas Debugger wurde komplett überarbeitet und sieht in der aktuellen Version folgendermaßen aus:

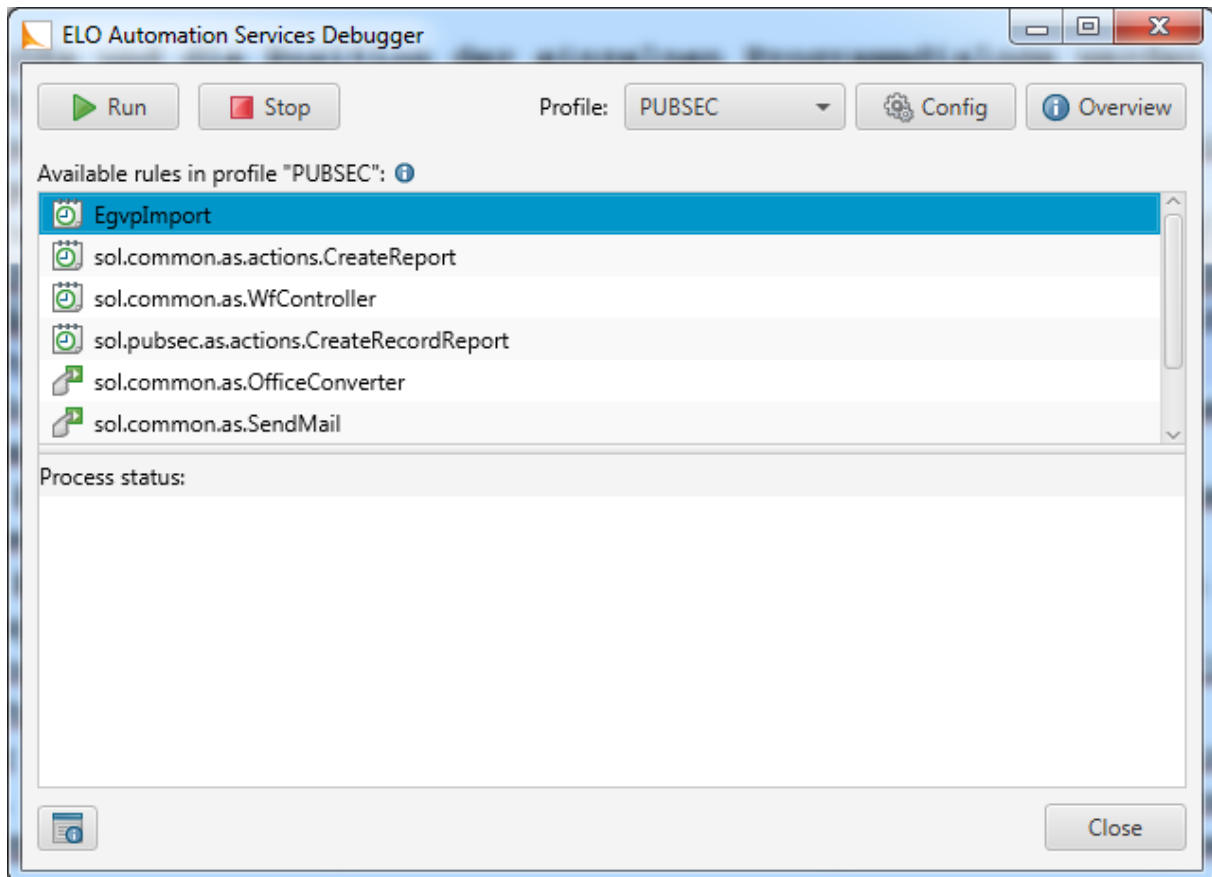


Abb. 1: Benutzeroberfläche des ELOas Debuggers

Nach dem Start des ELOas Debuggers ist standardmäßig immer die erste Regel ausgewählt.



Hinweis: Der ELOas Debugger ist derzeit nur auf Englisch verfügbar.

Die Größe und die Position der einzelnen Programmdialoge werden in der System-Registry gespeichert und beim nächsten Programmstart wiederhergestellt. Betroffen sind der Hauptdialog des ELOas Debuggers, der Dialog des Programms *LogFactor5*, der Dialog für den Rhino-Debugger, der Dialog für die Profilübersicht, für die Profil- und Parameterkonfiguration und der Dialog *About this program*.

Die Spaltenanordnung im Dialog *LogFactor5* wird ebenfalls in der System-Registry gespeichert und beim nächsten Programmstart wiederhergestellt.

Der ELOas Debugger enthält einen Splitter zwischen der Liste mit den vorhandenen Regelsätzen und dem Statusbereich. Die Splitter-Position wird in der System-Registry gespeichert und beim nächsten Programmstart wiederhergestellt.

1.1 Statusmeldungen

Im ELOas-Debugger-Hauptfenster ist ein mehrzeiliges Textfeld für die Statusmeldungen des Hintergrundprozesses vorhanden. Dieses Feld ist nicht editierbar, dessen Inhalt kann aber in die Zwischenablage kopiert werden.

1.2 Programminformationen

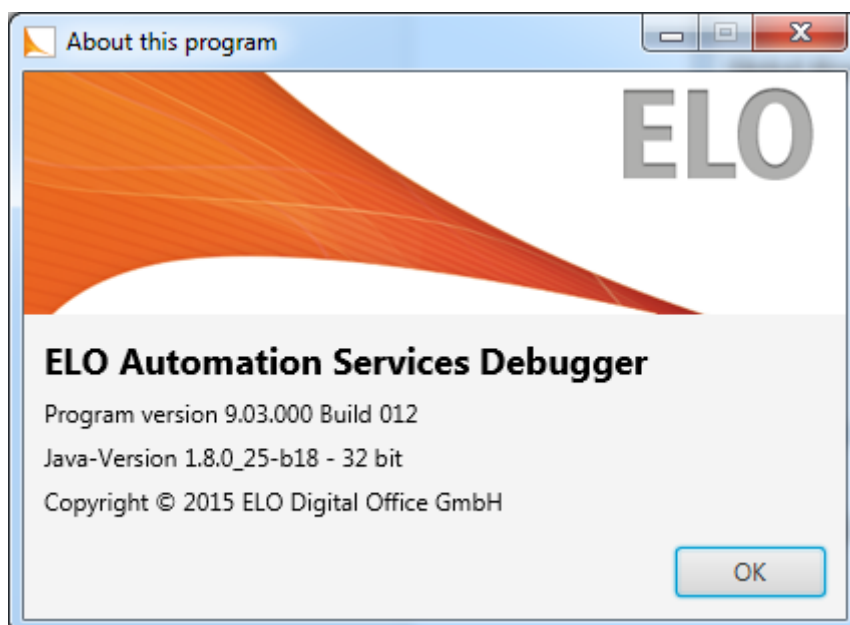


Abb. 2: Programminformationen

Im unteren linken Dialogteil ist die Schaltfläche für die Programminformationen platziert. Beim Betätigen dieser Schaltfläche wird der Dialog mit den Programminformationen angezeigt (Programmversion und verwendete Java-Version). Dieser Dialog kann über die OK-Schaltfläche oder über die Escape-Taste geschlossen werden.

2 ELOas Debugger starten

Mit Hilfe der Schaltfläche *Run* wird der Debug-Prozess einer AS-Regel gestartet. Dabei wird der Java Rhino ELOas Debugger in einem separaten Fenster angezeigt. Falls die Option für die LF5-Ausgabe im aktuell eingestellten ELOas-Debugger-Profil aktiviert ist, wird dieses Programm ebenfalls in einem separaten Fenster gestartet. In diesem Fenster werden die ELOas-Debugger-Ausgaben angezeigt. Die verschiedenen Logstufen können bei Bedarf mit einer passenden Farbe gekennzeichnet werden. Mit Hilfe der Schaltfläche *Stop* wird der Debug-Prozess einer AS-Regel abgebrochen. Die Schaltflächen für den Start und Abbruch einer Regel sind in der aktuellen ELOas-Debugger-Version immer aktiv.

Wenn keine AS-Regel ausgewählt ist und die Run-Schaltfläche betätigt wird, wird dem Benutzer der folgende Hinweis angezeigt.

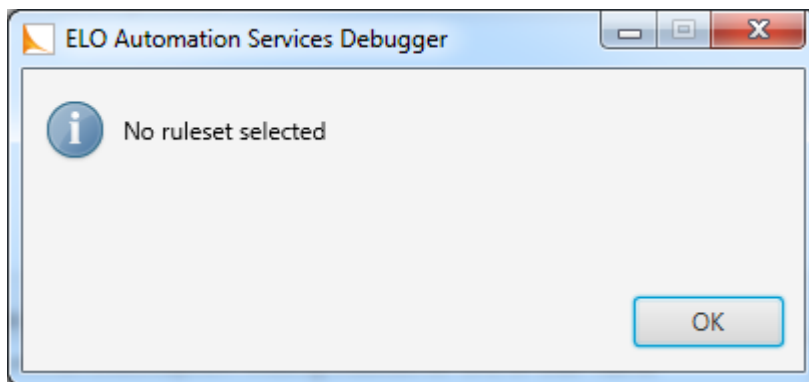


Abb. 3: Fehlermeldung bei Start ohne gewählte AS-Regel

3 Profile

3.1 ELOas Debugger-Profile verwenden

In der oberen rechten Dialogseite vom ELOas Debugger Hauptfenster ist eine Drop-down-Menü mit den ELOas Debugger-Profilen vorhanden. Dieses Drop-down-Menü enthält 10 Profile. Jedes Profil ist anhand des Profilnamens eindeutig identifizierbar.

Beim Auswählen eines Profils werden die betroffenen grafischen Dialogkomponenten aktualisiert. Die Liste mit den vorhandenen AS-Regeln wird bei Änderung des Profils auch aktualisiert. In dieser Liste werden die Regeln aus dem aktuell eingestellten ELOas Debugger-Profil angezeigt. Bei einem Doppelklick auf eine AS-Regel wird der Debug-Prozess für diese Regel gestartet. Mit einem rechten Mausklick auf eine Regel wird das Kontextmenü der Liste angezeigt. Im angezeigten Kontextmenü kann der Benutzer den Debug-Prozess für die ausgewählte AS-Regel starten oder abbrechen.

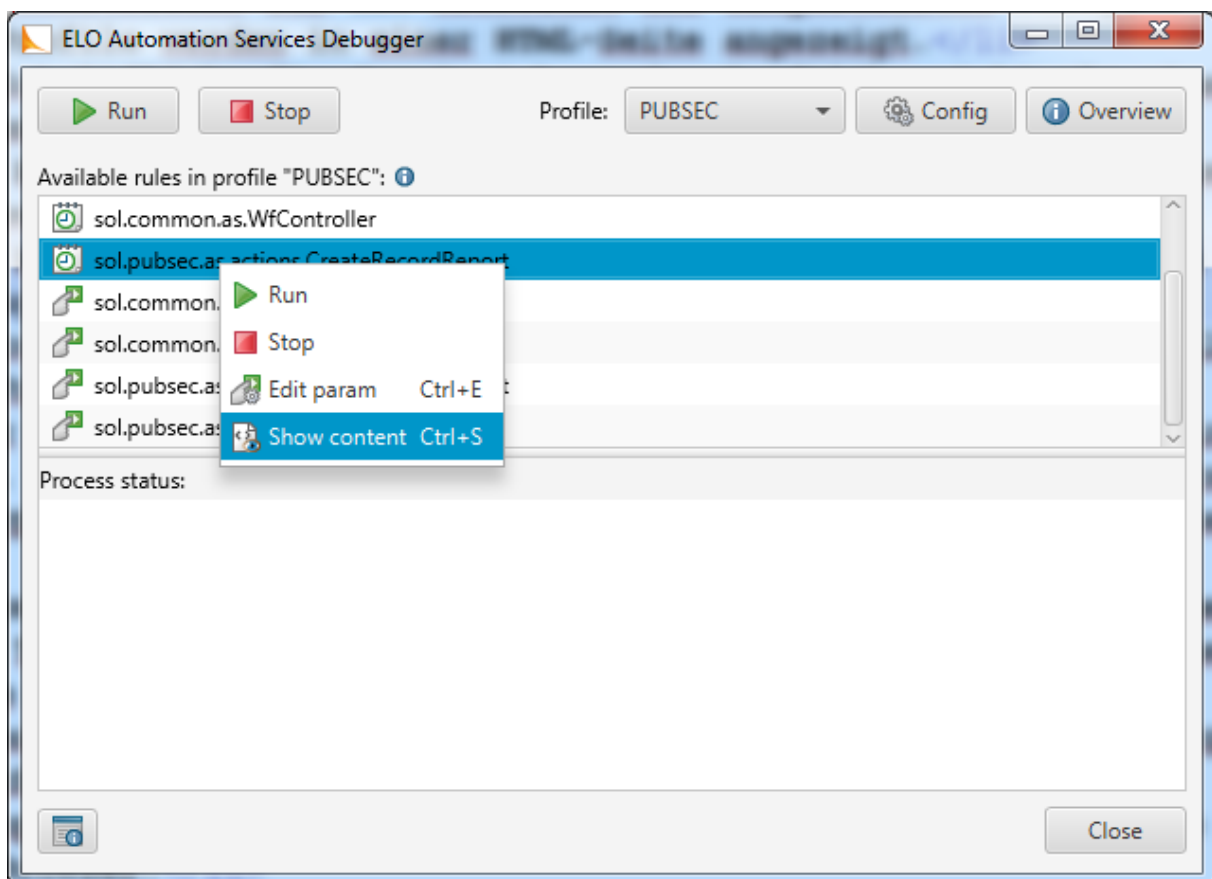


Abb. 4: Kontextmenü im ELOas Debugger

In der Liste mit den verfügbaren Regelsätzen werden nun auch die direkten Regelsätze angezeigt. Die einzelnen Regelsätze werden anhand ihres Symbols unterschieden. Das Typ-Symbol eines Regelsatzes enthält einen passenden Beschreibungstext.

3.2 Profilübersicht

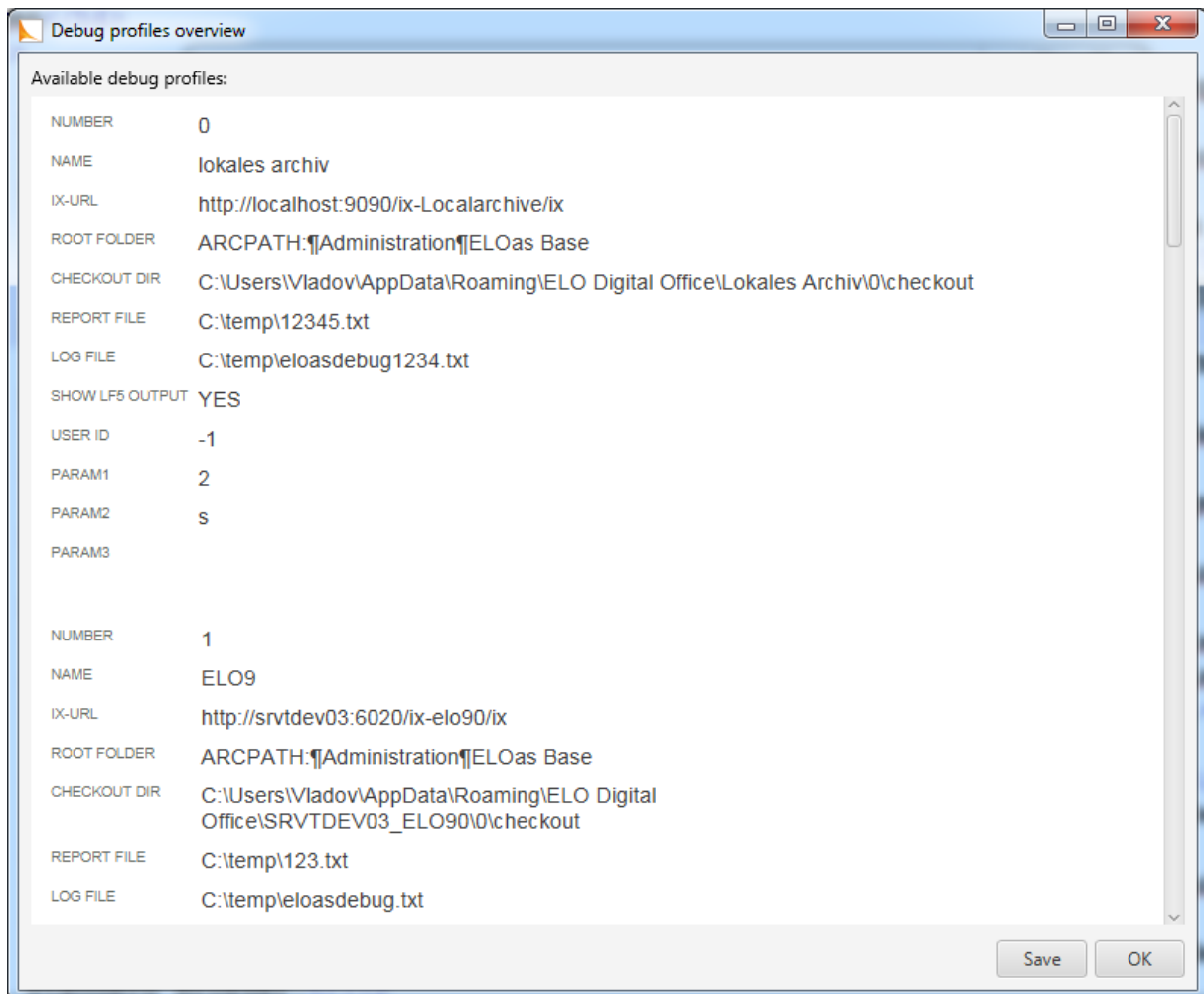


Abb. 5: Profilübersicht des ELOas Debuggers

Die Schaltfläche *Overview* öffnet die Übersicht der vorhandenen ELOas-Debugger-Profile. Die vorhandenen Profile werden in einer HTML-Seite angezeigt. In der Übersicht werden die wichtigen Eigenschaften eines Profils angezeigt. Die Profilübersicht kann über die Schaltfläche *Save* als HTML-Datei im lokalen Dateisystem gespeichert werden.

3.3 Informationen zum gewählten Profil

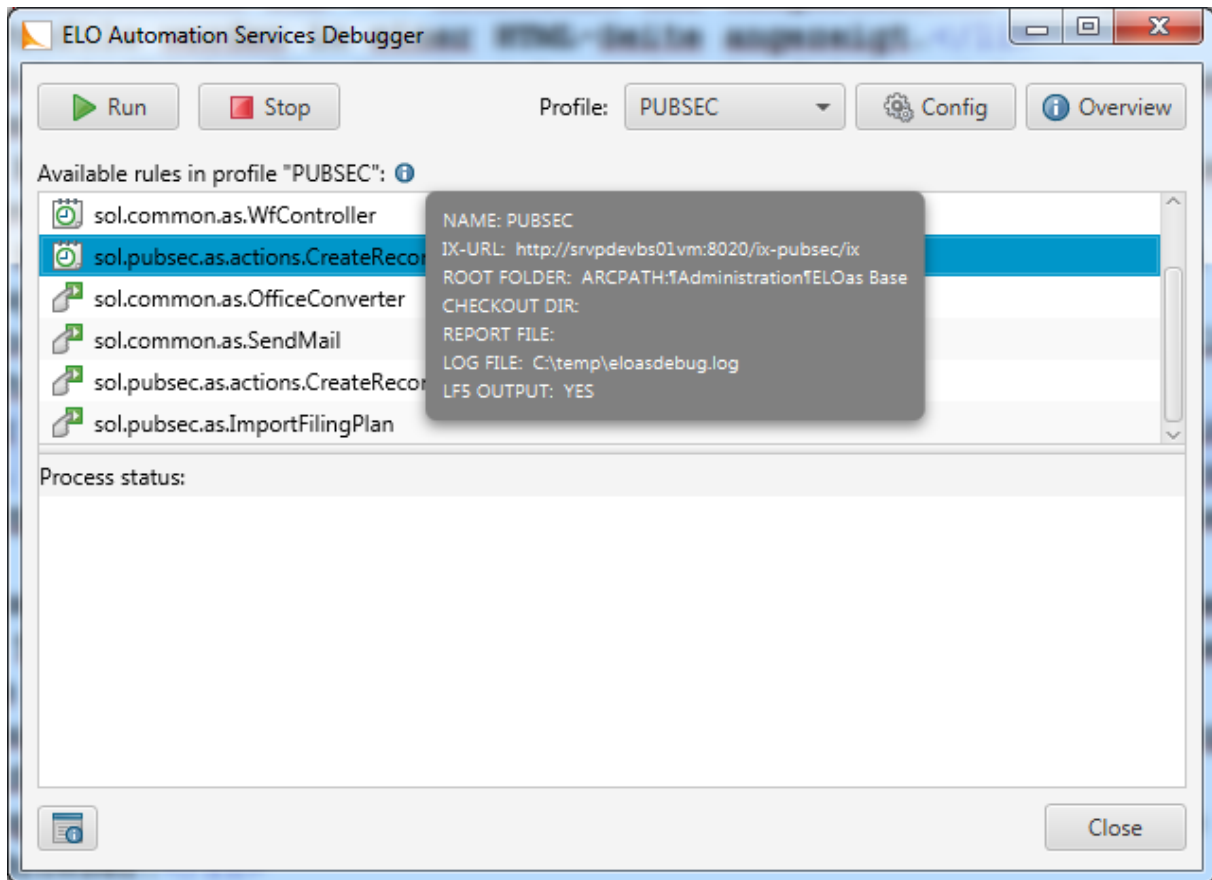


Abb. 6: Informationen zum gewählten Profil

Im Hauptdialog des ELOas Debuggers ist nun ein Info-Symbol mit den wichtigsten Daten des aktuell ausgewählten ELOas-Debugger-Profiles vorhanden.

3.4 Profile bearbeiten

Das Bearbeiten des aktiven ELOas-Debugger-Profiles ist über die Schaltfläche *Config* möglich.

Configuration of profile 3

Profile name

Name: PUBSEC

Indexserver connection

ELO user: ELO Service

Password: ...

IX-URL: http://srvpdevbs01vm:8020/ix-pubsec/ix

Automation Services configuration

Root folder: ARCPATH:AdministrationELOas Base

Local client configuration

Checkout dir: [Folder icon]

Report file: [Folder icon]

Log file: C:\temp\eloasdebug.log [Folder icon]

☒ Show LF5 output

Global direct rule parameters

User ID: -1

Param1: 1

Param2: {"tempObjId": "5119"}

Param3: [Empty field]

OK Cancel

Abb. 7: Dialog zum Konfigurieren eines Profils

Im Dialogtitel wird die Id des zu bearbeitenden ELOas-Debugger-Profiles angezeigt. Im Textfeld für das Benutzerpasswort wird das Benutzerpasswort nun maskiert.

Name: Der Profilname darf maximal 15 Zeichen lang sein.

IX-URL: Das Textfeld bekommt eine grüne Hintergrundfarbe wenn eine Verbindung möglich ist.

Checkout dir: Die Schaltfläche hinter dem Feld *Checkout dir* ermöglicht die Auswahl des Checkout-Verzeichnisses mit Hilfe des Dateiauswahldialogs von Java.

Report file: Über die Schaltfläche hinter dem Feld *Report file* lässt sich eine ELOas-Reportdatei auswählen.

Log file: Die Schaltfläche hinter dem Feld *Log file* ermöglicht die Auswahl der Logdatei mit Hilfe des Dateiauswahldialogs von Java.

Der Profilkonfigurationsdialog hat einen Scroll-Balken, der beim Verkleinern des Dialogs ab einer bestimmten Dialoggröße angezeigt wird.

Nach dem Betätigen der OK-Schaltfläche werden die durchgeführten Profil-Einstellungen in der System-Registry gespeichert. Die Einstellungen für das aktuelle ELOas-Debugger-Profil (Id: 1) werden an der folgende Position in der System-Registry gespeichert: „HKEY_CURRENT_USER\Software\JavaSoft\Prefs\elo digital office\eloas.1“.

Über die Schaltfläche *Abbrechen* werden die aktuellen Änderungen verworfen und der Dialog wird geschlossen. Der Dialog zum Bearbeiten eines ELOas-Debugger-Profiles kann auch über die Escape-Taste geschlossen werden. Dieser Dialog hat auch eine passende minimale Dialoggröße. Beim Vergrößern des Dialogs werden die einzelnen Dialogkomponenten ebenfalls passend vergrößert. Somit kann der Benutzer lange Profileingaben sehen.

3.5 Bearbeiten von direkten Regeln

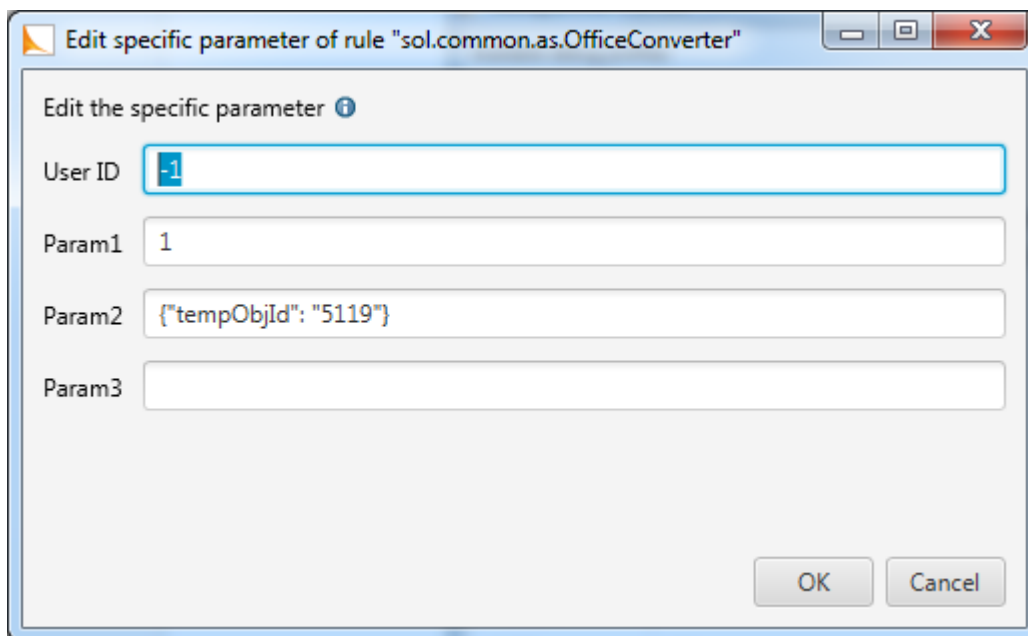


Abb. 8: Dialog zum Bearbeiten von direkten Regeln

Das Kontextmenü der Liste mit den vorhandenen Regeln wurde um einen Menüeintrag für das Bearbeiten der Parameter einer direkten Regel erweitert. Im angezeigten Dialog können die Benutzer-ID und die drei vorgesehenen Parameter bearbeitet werden.

4 Tastaturkombinationen

Die einzelnen ELOas-Debugger-Dialoge sind auch über Tastaturkombinationen aufrufbar.

F1 zeigt den Dialog *About this program* an.

STRG+P zeigt den Dialog für die Profilkonfiguration an.

STRG+O öffnet die Profilübersicht.

STRG+E zeigt den Dialog zum Bearbeiten der Parameter an.

STRG+S bringt den Skriptinhalt im Vordergrund.

STRG+L öffnet die Logdatei zur Ansicht.

STRG+R öffnet die eingestellte Reportdatei zur Ansicht.

5 Java-Bibliotheken

Das folgende Kapitel enthält eine Liste mit den Java-Bibliotheken, die im ELOas Debugger enthalten sind.

Nr.	Bibliothek	Beschreibung
1.	aspose-words-14.9.0.jdk16.jar	Bibliothek für das Erstellen und Verwalten von Microsoft-Word-Dokumenten.
2.	EloixClient.jar	Bibliothek für den Zugriff auf den Index Server (Index Server Schnittstelle).
3.	bcprov-jdk16-146.jar	Bibliothek für den Umgang mit verschlüsselten Dokumenten.
4.	eloserverutils.jar	Allgemeine ELO-Server-Hilfsklassen.
5.	commons-lang-2.6.jar	Hilfsmethoden für String-Operationen, Serialisierung und Objekt Reflection.
6.	commons-logging-1.2.jar	Bibliothek für Logausgaben in Java-Anwendungen.
7.	dom4j-1.6.1.jar	Flexible und freie Bibliothek für die Arbeit mit XML-Dateien, XPath-Ausdrücken und XSLT-Dateien.
8.	aspose-cells-8.2.1.jar	Bibliothek für das Erstellen und Verwalten von Microsoft-Excel-Dokumenten.
9.	aspose.slides-14.7.0.jar	Bibliothek für das Erstellen und Verwalten von Microsoft-Power-Point-Dokumenten.
10.	aspose-email-4.6.0.0-jdk16.jar	Bibliothek für das Lesen von E-Mail-Nachrichten
11.	httpclient-4.4.jar	Bibliothek für das Senden von http-Anfragen.
12.	httpcore-4.4.jar	Bibliothek für das Senden von http-Anfragen.
13.	jai_codec.jar	Bibliothek für die Bildbearbeitung in Java.

Nr.	Bibliothek	Beschreibung
14.	jai_core.jar	Weitere Bibliothek für die Bildbearbeitung in Java.
15.	jai_imageio patch.jar	Bibliothek für die Bildbearbeitung in Java.
16.	log4j-1.2.17.jar	Bibliothek für Logausgaben in Java-Anwendungen.
17.	js.jar	Bibliothek für den JavaScript-ELOas Debugger.
18.	commons-io-2.4.jar	Bibliothek mit nützlichen Hilfsmethoden für häufige Dateioperationen.
19.	mlibwrapper_jai.jar	Weitere Bibliothek für die Bildverarbeitung in Java.
20.	jna.jar	Bibliothek für den Zugriff auf System-Ressourcen.
21.	platform.jar	Weitere Bibliothek für den Zugriff auf System-Ressourcen.
22.	xmpbox-1.8.8.jar	Weitere Bibliothek für die Arbeit mit PDF-Dokumenten.
23.	stax2-api-3.0.2.jar	Bibliothek für eine erweiterte Standard-API für die Bearbeitung von XML-Dateien.
24.	woodstox-core-asl-4.1.1.jar	Bibliothek für die schnelle Bearbeitung von XML-Dateien.
25.	javautils.jar	Allgemeine ELO Java Utils.
26.	commons-codec-1.9.jar	Bibliothek mit allgemeinen Kodierer/Dekodierer-Klassen für Base64, Hex und URLs.
27.	fontbox-1.8.8.jar	Bibliothek für die Schriftart von PDF-Dateien.
28.	pdfbox-1.8.8.jar	Bibliothek für den Zugriff auf PDF-Dateien.

Nr.	Bibliothek	Beschreibung
29.	mail.jar	Bibliothek für den Zugriff auf eml-Dateien.
30.	xmpcore.jar	Bibliothek für das Bearbeiten, Drucken und Konvertieren von Dokumenten.
31.	metadata-extractor-2.6.4.jar	Bibliothek für das Extrahieren von Metadaten.
32.	json.jar	Bibliothek für die Erstellung von JSON-Strings.
33.	db2jcc.jar	Bibliothek für den DB2-Datenbank-Treiber.
34.	sqljdbc4.jar	Bibliothek für den Microsoft-SQL-Server-Datenbank-Treiber
35.	ojdbc6.jar	Bibliothek für den Oracle-Datenbank-Treiber.
36.	imgscalr	Bibliothek für die Bildskalierung
37.	poi-3.8-20120326.jar	Bibliothek für den Zugriff auf Office-Dokumente
38.	poi-scratchpad-3.8-20120326.jar	Weitere Bibliothek für den Zugriff auf Office-Dokumente
39.	core-3.1.0.jar	Bibliothek für die Barcode-Erkennung
40.	javase-3.1.0.jar	Weitere Bibliothek für die Barcode-Erkennung
41.	fop.jar	Bibliothek für die Konvertierung von XML-Dateien nach PDF
42.	xmlgraphics-commons-1.5.jar	Bibliothek für die Bearbeitung von XML-Dateien
43.	batik-all-1.7.jar	Bibliothek für Anwendungen, die Bilder in SVG-Format verwenden
44.	avalon-framework-4.2.0.jar	Bibliothek für die Erstellung und Konfiguration von Komponenten
45.	serializer-2.7.0.jar	Bibliothek für die Serialisierung

Nr.	Bibliothek	Beschreibung
46.	xalan-2.7.0.jar	Bibliothek für die Konvertierung von XML-Dokumenten nach HTML
47.	xercesImpl-2.7.1.jar	Bibliothek für einen XML-Parser
48.	xml-apis-1.3.04.jar	Java API für XML-Operationen
49.	xml-apis-ext-1.3.04.jar	Bibliothek für eine DOM, SAX, und JAXP Schnittstelle
50.	jsch-0.1.53.jar	Bibliothek für die Java Implementierung von SSH2
51.	jacob.jar	Bibliothek für den Zugriff auf COM-Objekte aus einer Java Anwendung
52.	Jacob-1.17-M2-x64-Lib.jar	Bibliothek mit den Jacob-DLLs für den Zugriff auf COM-Objekte aus einer Java Anwendung