

RICS Commander

Zusammenfassung

Diese Dokumentation beschreibt die Benutzung des RICS Commander und die API zum Checkservice.

Die Dokumentation darf nur mit Erlaubnis der infoGrips GmbH vervielfältigt werden.

Inhaltsverzeichnis

1. Einleitung	4
1.1. Aufbau dieser Dokumentation	4
1.2. Konventionen	4
2. Nutzung für Endanwender	5
2.1. Einleitung	5
2.2. Nutzung	5
2.2.1. Allgemeines	5
2.2.2. Parameter	5
2.2.3. Beispielaufrufe	5
3. API	6
3.1. Einleitung	6
3.2. API	6
3.2.1. Allgemeines	6
3.2.2. Verfügbare Methoden	6
3.2.3. Beispielanwendung	6

1. Einleitung

Mit dem Checkservice Commander (RICSC) lassen sich Dateien an den Checkservice schicken und Informationen zu den gesendeten Daten auslesen.

1.1. Aufbau dieser Dokumentation

Diese Dokumentation enthält:

- In Kapitel 1 einen Überblick und die in dieser Dokumentation verwendeten Konventionen.
- In Kapitel 2 die Beschreibung der Nutzung für Endanwender.
- In Kapitel 3 die Beschreibung der API.

1.2. Konventionen

In dieser Dokumentation werden folgende Konventionen eingehalten:

<i>Kursiv</i>	Namen von Dateien, wichtige Anmerkungen.
fett	neue Begriffe, Namen von Funktionen oder Methoden.
<code>courier</code>	Programmtext oder Eingaben im Betriebssystem.

2. Nutzung für Endanwender

2.1. Einleitung

In diesem Kapitel ist die Benutzung für den Endanwender beschrieben.

2.2. Nutzung

2.2.1. Allgemeines

Der RICS Commander kann über folgenden Befehl gestartet werden.

```
!CMD or Terminal
cd %ricsc_dir%
java -jar ricsc.jar %argumente%
```

2.2.2. Parameter

Folgende Parameter können in der Map RMDATAIN_PARAM für das Modul RMDATAIN gesetzt werden:

Parameter	Beschreibung
service	Servicename des Checkservice (zum Beispiel: checkch)
user	Benutzername (E-Mail-Adresse)
command	Kommando (send, check_level, get_log)
file	Relativer oder absoluter Pfad zur Datei. Beim Kommando "send" kann auch ein File-Pattern angegeben werden.
parameter-name	Je nach Checkservice lassen sich zusätzliche Parameter setzen. Diese können Einfach über die Kommandozeile eingegeben werden. (Beispiel: data_forward ON)

2.2.3. Beispielaufufe

```
!Datei an den Checkservice senden
java -jar ricsc.jar -service <service-name> -user <user-name> -command send -file *.xtf -data_forward ON

!Den Checklevel zu einer gesendeten Datei abrufen
java -jar ricsc.jar -service <service-name> -user <user-name> -command check_level -file test.xtf

!Den Link zur Log-Datei abrufen
java -jar ricsc.jar -service <service-name> -user <user-name> -command get_log -file test.xtf
```

3. API

3.1. Einleitung

In diesem Kapitel ist die API für den Zugriff zum Checkservice beschrieben.

3.2. API

3.2.1. Allgemeines

Mit der Java API lassen sich Checkservice-Aufgaben einfach in bestehende Umgebungen / Dienste einbetten.

Hierzu kann entweder der Source-Code oder das JAR-File in ein Projekt eingebunden werden.

3.2.2. Verfügbare Methoden

Prozedur	<code>public void ch.infogrips.service.rics.client.connect(String server, String usr, String passwd)</code>
Beschreibung	Stellt eine Verbindung zum Checkservice her.
Prozedur	<code>public void ch.infogrips.service.rics.client.disconnect()</code>
Beschreibung	Schliesst die Verbindung zum Checkservice.
Prozedur	<code>public void ch.infogrips.service.rics.client.sendFile(String fname, byte[] buffer, Hashtable options)</code>
Beschreibung	Sendet eine Datei mit den entsprechenden Optionen an den Checkservice.
Prozedur	<code>public Vector ch.infogrips.service.rics.client.getInfo(String service, String md5)</code>
Beschreibung	Ruft Informationen zu einer bestimmten Datei vom Checkservice ab.

3.2.3. Beispielanwendung

Als Beispielanwendung dient der RICS-Commander selbst. Die Klasse **RICSCCommander.java** ist der Main-Entry-Point.