NetWorker REST-API-Tools

Intelligente Skripte zur Erweiterung und Verbesserung der NetWorker V9 Bedienfreundlichkeit.

2018-02-14

NetWorker REST-API-Tools : Intelligente Skripte zur Erweiterung und Verbesserung der NetWorker V9 Bedienfreundlichkeit.

2018-02-14

1.1.1

Inhaltsverzeichnis

Was sind die NetWorker REST-API-Tools?

1. Einleitung

Seid der Version 9.0 besitzt NetWorker die neuen Ressource-Elemente Policys und Workflows, sowie seid der NetWorker Version 9.1 eine REST-API-Schnittstelle zur Verwaltung und Bearbeitung der NetWorker Ressourcen.

Leider fehlen bis Dato aber Mechanismen, die in der NetWorker Version 8.x im Umfeld der "alten" NetWorker Gruppen vorhanden waren. Dies sind zum Beispiel:

1. Der Start eines Workflows aus einem Batch-Job auf der Datenbank-Maschine, oder aus einem externen Scheduler heraus.

2. Die Ausgabe des "savegroup" Probe-Starts am Terminal.

3. Die Zusammenfassung aller Meldungen einer Workflow-Sicherung in einer Datei.

4. Die Emulation einer alten "savegroup"-Meldung, als Eingabe für bestehende Überwachungsmechanismen.

5. Die Filterung der Sicherungsmeldungen nach definierbaren Filterregeln (SCM-Filter).

NetWorker 9 besitzt mit der REST-API Schnittstelle die Möglichkeit die fehlenden Funktionalitäten selbst zu entwickeln. Hieraus entstand die Idee für die Erstellung der hier beschriebenen Tools. Diese gehen aber bereits heute über die oben erwähnten Funktionalitäten hinaus und es gibt bereits einige weitere Ideen auf der Roadmap.

2. Highlights

2.1. Zusammenfassung aller Meldungen eines Workflow-Laufes

nsr_workflow_analyse

Das Kommando erwartet als Eingabe die Notification-Ausgabe eines NetWorker Workflow-Laufes (siehe nsr_workflow_analyse) und ermittelt mit Hilfe des NSR-REST-API die zugehörigen Log- und Raw-Dateien. Diese werden aufbereitet und in einer Datei im WKF_LOG_BASE_DIR Verzeichnis mit dem Namen der Policy-Workflow Kombination abgelegt.

Nach der Generation einer zusammenhängenden Protokoll-Datei werden die generierten Sicherungsmeldungen mit Hilfe von definierbaren regulären Ausdrücken bewertet. Hierbei kann jede zu erwartende Meldung mit regulären Ausdrücken definiert und mit einem gewünschten Exit-Wert versehen werden. Die Exit-Werte bestimmen im Anschluß, was nach der Kompilierung der Meldungen mit dem Resultat passieren soll. Wenn kein Fehler in den Ausgaben gefunden wurde, soll nichts passieren (die Protokolle existieren ja sowieso). Im Falle eines minderwertigen Fehlers soll eine Nachricht an die Operateure gesendet werden. Bei schwerwiegenden Fehlern wird zusätzlich eine Nachricnt an die Administratoren gesendet. Sollte eine Meldung bisher unbekannt sein, so wird diese Information nur an die Administratoren gesendet!

Selbst wenn eine Fehlermeldung so nicht erwartet wurde, bekommen die Administratoren diese Information auf alle Fälle mitgeteilt.

Das Ziel ist also die Spreu vom Weizen zu trennen, bzw. die Administratoren für die gesendeten Meldungen wieder aufnahmefähig zu halten. Es macht schließlich keinen Sinn, täglich 1000 zeilige Meldungen zu erhalten in denen vielleicht an jedem 10'ten Tag eine fehlerhafte oder zu kontrollierende Meldung enthalten ist. Wenn diese eine Meldung aber als einzige markierte Meldung in einer Mail

übrig bleibt, da die 999 unbedenklichen Zeilen eleminiert wurden, sind die Sinne des Administrator geschärft und unerwartete Probleme werden leichter erkannt.

Als Alternative zur SCM-Ausgabe kann das Programm auch dazu verwendet werden, eine Emulation der "alten" NetWorker 8 Savegroup-Ausgabe durchzuführen.

In dem Falle wird eine Mail mit dem zugehörigen Betreff und Kopfbereich erzeugt, die der NetWorker 8 Savegroup-Ausgabe entspricht. Zusätzlich zum NetWorker 8 Kopf besteht der Inhalt der Mail aus dem Ergebnis der oben beschriebenen Filterung.

2.2. Intelligentes aktives und interaktives Start-Skript

start_workflow

Das Kommando kann interaktiv verwendet werden um anhand einer Auflistung der Policys, Workflows und deren zugehörigen Clients, gezielt einen oder mehrere Clients oder ganze Workflows zu starten. Hierdurch wird das Nachstarten einer Client-Sicherung mit einem definierten Backup-Level wesentlich vereinfacht.

Es ist auch möglich, alle gewünschten Parameter für den Start eines Workflows als Optionen mitzugeben, um diesen Start direkt in einen Batch-Job integrieren zu können.

Ein besonderes Highlight ist die Terminal-Ausgabe des Probe-Ergebnisses des Client-Kommandos **savefs -p**. Hiermit wird das Debugging von Start-Problemen eines NetWorker Clients wesentlich vereinfacht.

Kapitel I. Installation und Konfiguration

V1.1.1

I.1. Installation

I.1.1. Python "schtob" Library

Die NetWorker NSR-REST-API Tools verwenden zur Kommunikation mit dem NetWorker Diensten eine Schäfer & Tobies eigene Python-Library. Diese kann mit Hilfe des Shell-Skriptes **install.sh** in die Python Umgebung integriert werden.

Die Installation ist denkbar einfach. Starten sie das Skript nach dem entpacken des Tar/ZIP Files mit dem Kommando:

I.1.1.1. LINUX

```
python install.sh
```

I.1.1.2. Windows

XXX

I.1.2. Die NSR-REST-API-Tools

Die Ablage der NetWorker-REST-API Tools (Python Skripte) ist pfadunabhängig. Sie können sie ganz nach belieben im standard LINUX-Pfad zum Beispiel im Verzeichnis "/usr/local/bin" oder in einem eigenen Verzeichnis im NetWorker-Baum (zum Beispiel "/nsr/scripts/REST-API") ablegen.

I.2. Konfiguration

I.2.1. Kopieren der nsr_api_local_defines.py.template Datei

Für die erste Konfiguration der Datei **nsr_api_local_defines.py** gibt es in dem Python **schtob** Verzeichnis die Template-Datei **nsr_api_local_defines.py.template**. Diese sollten sie auf den Namen **nsr_api_local_defines.py** kopieren und anschließend die folgenden Parameter an ihre Umgebung anpassen.

Sollten sie nicht wissen, wohin das schtob-Python-Modul installiert wurde, empfieht sich folgendes Vorgehen:

```
# echo "import schtob
schtob
" | python -i
```

Die erhaltene Ausgabe dürfte in etwa folgendermaßen aussehen:

```
Python 2.7.6 (default, Nov 23 2017, 15:49:48)
[GCC 4.8.4] on linux2
Type "help", "copyright", "credits" or "license" for more information
>>> >>> >>> <module \'schtob' from '/usr/lib/python2.7/site-packages/
>>> >>>
```

In diesem Fall finden sie die Template-Datei im Verzeichnis **'/usr/lib/python2.7/site-packages/schtob'**. Die kopierte Datei sollte im selben Verzeichnis oder in einem beliebigen anderen Verzeichnis ihres Standard-Python-Pfades abgelegt werden.

I.2.2. Edieren der nsr_api_local_defines.py Datei

In der kopierten Template-Datei finden sie die unten aufgeführten Parameter, die sie für ihre Umgebung und Wünsche anpassen müssen bzw. können.

Die meisten Parameter sind auskommentiert (das erste Zeichen der Zeile ist ein '#'). Die nach dem Gleichheitszeichen angegebenen Werte geben in dem Fall den Default-Wert an, den sie durch das auskommentieren der Zeile auf einen für ihre Umgebung passenden Wert setzen können. Einige Parameter sind jedoch nicht kommentiert. Diese müssen von ihnen auf passende Werte gesetzt werden.

I.2.2.1. Allgemeine NSR_REST-API Parameter

Für die Verwendung der NSR-REST-API Tools, die direkt auf die API Schnittstelle zugreifen möchten, empfiehlt es sich die folgenden Parameter für die NSR-REST-API Kommunikation zu definieren. Weitere Informationen zur eigenen Verwendung der NSR-REST-API Library und deren Verwendung findet man im Kapitel schtob REST-API Library.

Für die Kommunikation mit dem NetWorker-Server benötigen die Tools einen gültigen NetWorker-Benutzer mit den ausreichenden Rechten. Diese können zwar bei dem Skript **start_workflow** als Optionen mitgegeben werden, aber für das Skript **nsr_workflow_analyse** müssen sie in der Datei **nsr_api_local_defines.py** definiert werden.

- **NSR_SERVER**

  Der Default-NetWorker-Server-Name

- **REST_API_USER**

  Der Default-NetWorker-Benutzer

- **REST_API_PASSWORD**

  Das Passwort für den Default-NetWorker-Benutzer

- **API_SERVER_CONFIG**

  Das **nsr_workflow_analyse** Kommando benötigt für die Kommunikation mit den NetWorker-Diensten einen gültigen Administrator-Eintrag in dieser Konfigurations-Struktur.

  Das Tool **start_workflow** greift ebenfalls auf die hier definierten Werte zu, wenn beim Aufruf des Programms ein NetWorker-Server-Name als Option mitgegben wird.

  Die Definition dieses Dictionarys empfiehlt sich demnach auch bei der Verwendung des Tools auf einem externen Scheduler und bei der Bearbeitung von mehreren NetWorker-Servern.

  Beispiel:

```
API_SERVER_CONFIG = {
    'nsr9-121' : {
        'user'     : 'administrator',
        'password' : 'Admin_123'
    },
    'lin25' : {
        'user'     : 'administrator',
        'password' : 'Fritz_123'
    },
}
```

Die NSR-REST-API Tools versenden in Fehler-Situationen Mails an die definierten Administratoren. Hierzu benötigt sie einen Default Mail-Empfänger, sowie eventuell den **MAIL_FROM** Parameter.

- **MAIL_USER**

  Der Default Mail Empfänger.

- **MAIL_FROM**

  Der Mail-Sender (siehe **mailx -r from-addr**).

- **REST_API_LOG_DIR**

  Verzeichnis in dem für die NSR-REST-API Kommunikation ein Unterverzeichnis mit dem Namen *restapi* angelegt wird. In diesem Verzeichnis werden für die einzelnen Tools, entsprechend bezeichnete Log-Dateien erzeugt.

- **REST_API_LOG_SIZE**

  Bei jedem Start eines NSR-REST-API Tools wird die Größe der bestehenden Logdatei überprüft. Sollte diese größer sein als die definierte **REST_API_LOG_SIZE** wird die Datei rolliert.

- **REST_API_LOG_COUNT**

  Anzahl der Logdateien, die für das jeweilige Tool aufbewahrt werden sollen.

- **DBG_DIR**

  Verzeichnis in dem für das jeweilige Tool, Debug-Dateien erzeugt werden sollen. In diesem Verzeichnis werden für die einzelnen Tools, entsprechend benamte Debug-Dateien erzeugt.

I.2.2.2. nsr_workflow_analyse Parameter

Die folgenden Parameter sind Definitionen für das Workflow-Analyse Tool **nsr_workflow_analyse**.

- **WKF_LOG_BASE_DIR**

  Verzeichnis in dem die Protokolle und Filter-Ergebnisse des Analyse-Tools abgelegt werden.

- **WKF_LOG_RETENTION_IN_DAYS**

  Aufbewahrungszeitraum der Analyse-Logs; (Anzahl in Tagen)

- **WKF_FILTER_FILE**

  Absoluter Pfad der Workflow-Analyse Filter-Datei.

  In dieser Datei werden alle bekannten NetWorker Workflow-Meldungen mit einer Fehler-Wertigkeit versehen. Abhängig von dem höchsten aufgetretenen Wert wird am Ende der Analyse, das im Parameter **COMPILE_ERROR_COMMANDS** definierte Kommando mit dem Ergebnis der Analyse aufgerufen.

- **NSR_OPERATORS**

  Python-Liste von Mail-Adressaten, die definierte Workflow-Fehler und Probleme erhalten sollen (siehe COMPILE_ERROR_COMMANDS).

- **NSR_ADMINS**

  Python-Liste von Mail-Adressaten, die erkannte Workflow-Fehler und Probleme erhalten sollen (siehe COMPILE_ERROR_COMMANDS).

- **COMPILE_ERROR_COMMANDS**

  Definition der Kommandos und deren Optionen, die als Ergebnis der Workflow-Analyse gestartet werden sollen.

  Zurzeit stehen für die Weiterleitung der erkannten Sicherungs- und Cloning-Probleme zwei Kommandos zur Verfügung. Dies ist zum einen das Kommando 'MAIL' (Aufruf des LINUX-Kommandos **mailx**) und zum zweiten des Kommando 'LOGGER' (Aufruf des LINUX-Kommandos **logger**). In Zukunft sollen hier auch beliebige andere Kommandos aufgerufen werden können.

  Für das Kommando LOGGER kann für jede definierte Priorität ein Kommentar und die SYSLOG bzw. JOURNALD Priorität mitgegeben werden.

  Für das Kommando MAIL wird das Elemet **comment** mit dem Namen der Policy und des Workflows und der Priorität des erkannten Fehlers (Element **prio**) der Mail als Subject übergeben.

  subject="NetWorker Workflow MSG: *Policyname-Workflowname* : Error-Code: *PRIO: COMMENT*"

  Die Mail-Adressaten können entweder durch die Verwendung der Parameter **NSR_OPERATORS**, **NSR_ADMINS** oder auch durch die direkte Angabe einer Python-Liste definiert werden. Wenn kein Mail-Adressat angegeben wird, wird die Mail and den Default Mail-Benutzer (**MAIL_USER**) gesendet.

  Die im Template verwendeten Dictionary Elemente "1", "2", "3" und "*" entsprechen den definierten Fehler-Wertigkeiten der Filter-Datei (siehe WKF_FILTER_FILE).

- **CHECK_SCHEDULED_CLIENT_ATTR**

  Vor der NetWorker Version 9 wurden Clients, die mit Hilfe des Client-Ressource-Attributes **"scheduled backup"** von einer Sicherung ausgeschlossen wurden, in der Savegroup-Completion-Message aufgelistet. Diese Auflistung ist in der Version 9.x nicht mehr vorhanden.

  Das Tool **nsr_workflow_analyse** ermittelt bei jedem Workflow-Lauf, ob es NetWorker-Clients gibt, die das Attribut "unscheduled backup" gesetzt haben. Ist dies der Fall, wird die Fehler-Priorität des Workflow-Laufs auf die Priorität "2" gesetzt. Der Administrator kann diese Meldungen für definierte Clients jedoch abschalten (siehe UNSCHEDULED_CLIENT_WHITE_LIST).

- **UNSCHEDULED_CLIENT_WHITE_LIST**

  Absoluter Pfadname einer Ascii-Datei, in der die Namen von NetWorker-Clients enthalten sind, für die keine Fehlermeldungen generiert werden sollen.

  Sollte ein Client mit Hilfe des Client-Attributes "scheduled backup" für immer aus der Sicherung ausgeschlossen worden sein, weil er z.B. nicht mehr existiert, so kann die Generierung einer Fehler-Benachrichtigung ausgeschlossen werden, indem der Client in der hier definierten Datei eingetragen wird.

- **OLD_SGRP_MSG**

  Generiere eine Mail, die der NetWorker 8 Savegroup-Completion-Message entspricht. Diese Emulation der NetWorker 8 Mail-Ausgabe soll Kunden, die bisher eine automatische Auswertung des Savegroup-Headers vorgenommen hatten, einen leichteren Übergang auf die NetWorker 9 Umgebung ermöglichen.

I.2.2.3. start_workflow Parameter

Die folgenden Parameter sind Definitionen für das **start_workflow** Tool.

Das Tool **start_workflow** bietet die Möglichkeit eine Datenbanksicherung mit Hilfe der Optionen *--client <client>* und *--db_id <DB-ID>* zu starten. Diese Funktionalität ist hauptsächlich für Datenbank-Umgebungen gedacht, in denen mehrere Datenbanken auf einem Client laufen. Hierbei wird

davon ausgegangen, dass die zugehörige Client-Ressource eindeutig über das SaveSet Attribut des Clients gefunden werden kann.

- **DB_ID_SPLIT_CHARS**

  Trennzeichen zwischen dem Datenbankmodul-Namen und der Datenbank-ID.

- **DB_MODULES**

  Liste von Datenbank-Modulnamen. Wenn keiner der hier definierten Modulnamen im SaveSet vorkommt, wird der Client nicht als Datenbank-Client betrachtet.

I.2.3. Kopieren der Workflow-Filter-Datei

Im Unterverzeichnis *schtob/nsr_tools* der installierten NSR-REST-API Tools, liegt die Template Datei *filter_msgs.template*. Diese muss auf den Pfadnamen des Parameters **WKF_FILTER_FILE** kopiert werden. Wenn dieser Parameter von ihnen nicht umdefiniert wurde, sucht das Filter-Tool die Datei mit dem Namen */nsr/res/wkf_filter_msgs.*

I.2.4. Edieren der Workflow-Filter-Datei

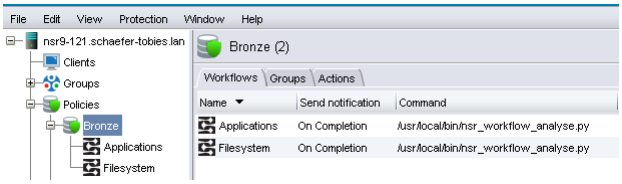
Das Tool **nsr_workflow_analyse** kann für dieselbe Meldung unterschiedliche Exit-werte erzeugen und damit unterschiedliche Kommandos oder Mail-Adressaten bedienen. Hierzu müssen jedoch die im NetWorker definierten Protection_group-Namen der Workflows eingetragen werden.

Eine genaue Beschreibung der in dieser Datei einzutragenden Werte können dem Kapitel **WKF_FILTER_FILE** entnommen werden.

I.2.5. Aktivierung der Workflow-Analyse

Das Kommando **nsr_workflow_analyse** erwartet als Eingabe die Notification-Ausgabe eines Workflowlaufs.

Um das Tool zu aktivieren, genügt demnach das Setzen des Workflow-Notification-Commands.



Definition der Workflow-Notification Kommandos

Setzen sie bei allen Workflows das Attribut-Notification auf den absoluten Pfadnamen des **nsr_workflow_analyse** Kommandos, so werden für jeden Workflow-Lauf im Protokoll-Verzeichnis 3 Dateien generiert.

Die unten aufgeführten Beispieldateien gehören zu einem Workflowlauf der Policy "Bronze" mit dem Workflownamen "fs1" und der Startzeit "2017.12.15 11:18:32".

- Eine Kopie der Workflow-Ausgabe (z.B. *Bronze_fs1_2017_12_15_11_18_32.out*)
- Eine Zusammenfassung aller durch den Workflowlauf erzeugten Protokolldateien. (z.B. *Bronze_fs1_2017_12_15_11_18_32__458376.log*)
- Das Ergebnis des Filter-Laufs über die Workflow-Ausgabe. (z.B. *Bronze_fs1_2017_12_15_11_18_32_flt_msg.txt*)

Kapitel II. Das nsr_workflow_analyse Kommando

V1.1.1

II.1. Einleitung

Das nsr_workflow_analyse Kommando ist als Ersatz für das ehemalige Savegroup-Completion-Message (SCM) Tool von Fujitsu-Siemens entwickelt worden. D.h. das Kommando soll die von den NetWorker Sicherungs-Prozessen erzeugten Meldungen analysieren und das Ergebnis der Analyse an ein definiertes Kommando senden. Die Analyse erfolgt anhand einer Filter-Datei, die aus regulären Ausdrücken und definierten Exit-Werten besteht. Die jeweiligen Exit-Werte der Filter-Analyse bestimmen dann das auszuführende Komando.

Bei der Filter-Analyse wird jede Meldung gegen Reguläre-Ausdrücke kompiliert. Passt ein regulärer Ausdruck, so wird der für die zugehörige Workflow-Definition definierte Exit-Wert für die Meldung gesetzt. Die regulären Ausdrücke können hierbei auch mehrzeilig sein. Nach der Kompilierung aller Meldungen wird anhand des höchsten gefundenen Exit-Wertes entschieden, welche Folgeaktion ausgeführt werden soll. Hierbei kann entweder eine Mail an unterschiedliche Adressaten gesendet werden, das Filter-Ergebnis an den Syslog-Daemon weitergeleitet werden oder ein beliebiges anderes Kommando ausgeführt werden. In allen Fällen kann sowohl die NetWorker Notfication, sowie eine Zusammenfassung aller Workflow Meldungen und das Ergebnis des Kompilierlaufes im Protokollverzeichnis kontrolliert werden.

Sollte eine Zeile zu keinem regulären Ausdruck passen, so wird dies als schwerwiegender Fehler gewertet und dem Exit-Wert wird die Zahl 99 zugeordnet. Somit ist gewährleistet, dass keine unbekannte Fehlermeldung durch das Suchraster fallen kann.

II.2. Voraussetzungen

- Der Aufruf des Kommandos erfolgt auf einem NetWorker-Server.
- Die Login-Informationen für den lokalen NetWorker-Server sind in der Konfigurationsdatei **nsr_api_local_defines.py** in der Python-Struktur **API_SERVER_CONFIG** enthalten.
- Der Start des Kommandos erfolgt über den NetWorker Workflow Notification Aufruf.

  Das **nsr_workflow_analyse** Kommando benötigt für die Erzeugung eines kombinierten Workflow-Logs die Ausgabe einer NetWorker Workflow-Notification.

  Eine Workflow-Notification sieht in etwa folgendermaßen aus:

```
---Summary notification report---
Policy name:Filesystem
    Workflow name:fs1, Workflow status:succeeded, Workflow start time:12/15/17
        Action name:backup, Action status:succeeded, Action start time:12/15/1
        Action name:clone, Action status:succeeded, Action start time:12/15/17
```

- Die Job-Informationen der zum Workflow gehörigen Backup- und Clone-Jobs müssen noch in der Job-Datenbank enthalten sein.
- Die Workflow-Filter-Datei muss existieren.

  Der Default-Pfad für die Filter-Datei ist: */nsr/res/wkf_filter_msgs*. Der Dateiname kann mit Hilfe des Parameters WKF_FILTER_FILE in der *nsr_api_local_defines.py* Datei überdefiniert werden.
- Der Policy/Workflow Name muss in der Filter-Datei eingetragen sein.

Für eine korrekte Bewertung der Filter-Regeln muss der Workflow Name in der Filter-Datei als Workflow-Profil eingetragen sein. (sieh WKF_PROFIL Definitionen)

II.3. Die Filter-Datei

Eine Template-Filter-Datei findet man im Installationsverszeichnis der REST-API-Tools. Der genaue Pfad lautet:

<Installation-dir>/schtob/nsr_tools/filter_msgs.template

Sie sollten diese Template-Datei am besten auf den Namen */nsr/res/wkf_filter_msgs* kopieren und dann zunächst einmal die **WKF_PROFIL** Definitionen vornehmen.

II.3.1. WKF_PROFIL Definitionen

Eine Profil-Definition besteht aus der Definitionsbezeichnung **WKF_PROFIL**, dem Profilnamen und den in diesem Profil zusammengefassten Protection-Gruppen.

Beispiel:

```
WKF_PROFIL FILESYSTEM: Filesystem_PCs,Filesystem_Linux, Filesystem_Win
WKF_PROFIL SERVER:     "Server Protection/Server backup"
WKF_PROFIL NMC:        "Server Protection/NMC server"
WKF_PROFIL DB:         "Datenbanken_SAPHANA"
WKF_PROFIL VMWARE:     "VMware/Aachen_hotadd","VMware/nwvcs_hotadd"
WKF_PROFIL NDMP:       "NDMP_Workflow1","NDMP_Workflow2","NDMP_Workflow3"
WKF_PROFIL CLONE:      "Clone/Clone_filesystem"
```

Der Profilname und die Gruppennamen müssen aus syntaktischen Gründen durch das Zeichen ":" voneinander getrennt werden. Mehrere Gruppennamen werden durch ein ',' voneinander getrennt. Wenn der Gruppenname Leerzeichen enthalten sollte, so muss der Gruppenname in Hochkommas '"' eingeschlossen werden.

II.3.2. Die Filter-Regeln

Die Filterregeln bestehen aus einzeiligen oder mehrzeiligen regulären Ausdrücken, gefolgt von einer Exit-Definition. Die Exit-Definition kann entweder aus einem Default-Wert (DEF=<nr>) oder aus mehreren Exit-Werten für unterschiedliche Profile und dem Default-Wert bestehen.

Beispiele:

```
"^\* .*  nsrworkflow Starting Protection Policy .* workflow.*"
       DEF=0

"^\* .*  nsrworkflow Starting action.*with command:"
"^\* .*  nsrworkflow Action .*log will be in"
       DEF=0
```

Der erste reguläre Ausdruck definiert eine NetWorker Workflow Start-Information. Diese Meldung ist in keinem Zusammenhang als Fehler zu bewerten, daher ist der Exit-Wert für diesen regulären Ausdruck für alle Workflows auf den Wert 0 vordefiniert. Der zweite reguläre Ausdruck besteht aus 2 Zeilen, die direkt hintereinander in der Ausgabe des Programms erscheinen müssen. Auch diese Meldungen sind in keinem Kontext als Fehler zu bewerten und erhalten daher als Default-Exit-Wert den Wert 0.

```
"^\* .*  nsrworkflow.*failed"
        DEF=3
```

Dieser reguläre Ausdruck ist auf alle Fälle ein Problem und sollte daher an die Administratoren gemeldet werden. Der Default-Exit-Wert 3 führt dazu, dass das in den **Informations-Kommandos** angegebene Kommando für den Exit-Wert 3, am Ende der Bearbeitung ausgeführt wird. Die durch diesen regulären Ausdruck gefundene Meldung wird auf alle Fälle in der zusammengefassten Filter-Ausgabe enthalten sein.

Damit das Tool für unterschiedliche Workflows, unterschiedliche Adressaten erreichen kann, können für einen regulären Ausdruck unterschiedliche Exit-Werte definiert werden. Dies kann man erreichen, indem man unterhalb des zugehörigen Regulären-Ausdrucks für unterschiedliche Profile unterschiedliche Exit-Werte definiert.

```
"^\* .*:save: Unable to setup direct save with server [^ ]*: current save sessi
        DEF=2
        FILESYSTEM=1
```

Meldungen die auf den obigen regulären Ausdruck passen, sollen bei Protection-Gruppen, die in dem Profil **FILESYSTEM** zusammgefasst wurden, mit dem Kommando des Exit-Wertes 1 bearbeitet werden. Sollte es sich um einen anderen Workflow handeln, so wird das **Informations-Kommando** des Exit-Wertes 2 gestartet.

II.4. Die Definition der Informations-Kommandos

Die Benachrichtigung der Administratoren über fehlgeschlagene Workflows kann durch unterschiedliche Messaging-Programme erfolgen. In den meisten Fällen werden wohl Mails an die NetWorker-Administratoren genügen, aber es können zum Beispiel für definierte Datenbank-Probleme auch zusätzlich Mails, an die Datenbank-Admins gesendet werden. Minderwertige Probleme sollen vielleicht auch nur protokolliert oder an die NetWorker-Operateure versendet werden.

Zurzeit werden nur die beiden Kommandos **MAIL** und **LOGGER** unterstützt.

Die Default-Definiton

```
COMPILE_ERROR_COMMANDS = {
    '1' : { 'cmd' : 'LOGGER',
            'comment'   : 'eine Warnung gefunden',
            'prio'      : 'NOTICE',
          },
    '2' : { 'cmd' : 'MAIL',
            'comment'   : 'minderwertiger Fehler',
            'prio'      : 'WARNING',
            # Mail geht an den 'normalen' MAIL_USER
          },
    '3' : { 'cmd' : 'MAIL',
            'comment'   : 'schwerwiegender Fehler',
            'MAIL_USER' : NSR_OPERATORS + NSR_ADMINS,
            'prio'      : 'ERROR',
          },
    '*' : { 'cmd' : 'MAIL',
            'comment'   : 'Not defined Message found',
            'MAIL_USER' : NSR_ADMINS,
            'prio'      : 'UNDEFINED',
          },
}
```

Die obige Definition führt zu folgenden Bearbeitungen der Kompilierungs-Ergebnisse:

Sollte der Workflow mit dem Exit-Wert 1 beendet worden sein, so wird das Ergebnis der Kompilierung an das Kommando **LOGGER** gesendet. Hierbei wird dem Syslog-Daemon bzw. dem Journal-Daemon die Meldungen mit der Priorität **NOTICE** übergeben.

Bei allen anderen Exit-Werten wird eine Mail versendet. Die Mails unterscheiden sich jedoch in den Adressaten und der Betreffszeile.

- Exit-Wert 2 führt zu der Versendung einer Mail, an die in der *nsr_api_local_defines.py* Datei definierten **MAIL_USER**.
- Exit Wert 3 erzeugt eine Mail an die definierten NetWorker Operateure und Administratoren (**NSR_OPERATORS** und **NSR_ADMINS**).
- Alle Exit-Werte größer als 3 erzeugen eine Mail an die definierten NetWorker Administratoren (**NSR_ADMINS**). Diese Aktion wird vor allem bei bisher unbekannten Meldungen ausgeführt.

Der Betreff der Mails wird wie folgt zusammengesetzt:

<NetWorker Server-Name>**: NetWorker Workflow MSG:**<Protection Group Name> **: Error-Code:** <Exit-Wert> **:** <comment>

Beispiel:

```
nsr9-121: NetWorker Workflow MSG: FS-Backup_Clone : Error-Code: 3: schwerwiegen
```

Es ist ihnen überlassen, diese Kommando Liste zu erweitern, oder die Kommentare in ihrem Sinne anzupassen.

II.5. das Client Attribut "scheduled backup"

In der NetWorker Version 8 wurden in den Savegroup Meldungen, Clients deren Sicherung mittels dem Attribut **"scheduled backup"** ausgeplant worden waren, in einer eigenen Zeile ausgegeben. Der Administrator wurde also immer daran erinnert einen Client, der nur vorübergehend ausgeplant wurde, (dafür ist das Attribut ja gedacht) wieder einzuplanen. Das Deaktivieren der Client-Sicherung funktioniert weiterhin, nur leider erhält der Administrator keinen Hinweis mehr darüber.

Das **nsr_workflow_analyse** Tool kontrolliert nach der Analyse der Workflow-Meldungen, ob es für den betreffenden Workflow Client-Definitionen gibt, bei denen das Attribut **scheduled backup** deaktiviert ist. Werden solche Clients gefunden, so werden diese in dem Ergebnis der Filter-Datei im Kopfbereich in einer eigenen Zeile, mit der Einleitung **WORKFLOW HAS UNSCHEDULED CLIENTS:** ausgegeben.

Der Admnistrator kann das Feature über den Konfigurationsparameter **CHECK_SCHEDULED_CLIENT_ATTR** deaktivieren.

Den Exit-Wert für dieses Ereignis kann man in der Filter-Datei setzen (siehe "UNSCHEDULED Clients found").

II.6. Protokolle und Debugging

II.6.1. Protokolle

II.6.1.1. Protokollverzeichnisse

Die NetWorker Notification-Ausgabe, die Zusammenfassung aller Workflow-Meldungen, sowie das Ergebnis der Kompilierung werden in einem tagesspezifischen Verzeichnis unterhalb des Workflowlog-Verzeichnisses abgelegt. Das Workflowlog-Verzeichnis befindet sich per Default im NetWorker-Log Verzeichnis (*/nsr/logs/workflow_logs*. Der Pfad des Log-Verzeichnisses kann jedoch mit Hilfe des Parameters **WKF_LOG_BASE_DIR** in der *nsr_api_local_defines.py* Datei geändert werden.

Unterhalb des Workflowlog-Verzeichnisses findet man folgende Verzeichnisstrucktur:

```
<Jahr>/<Monat>/<Tag>
```

Beispiel:

```
# ls -R
.:
2018

./2018:
01

./2018/01:
10  11  12  13 14 15 ... 30 31

./2018/01/10:
FS_Backup_Clone_2018_01_10_13_35_00__128151.log.gz
FS_Backup_Clone_2018_01_10_13_35_00_flt_msg.txt
FS_Backup_Clone_2018_01_10_13_35_00.out
Server Protection_NMC server backup_2018_01_10_14_00_00__128178.log.gz
Server Protection_NMC server backup_2018_01_10_14_00_00_flt_msg.txt
Server Protection_NMC server backup_2018_01_10_14_00_00.out

./2018/01/12:
FS_Backup_Clone_2018_01_12_13_35_00__192016.log.gz
FS_Backup_Clone_2018_01_12_13_35_00_flt_msg.txt
FS_Backup_Clone_2018_01_12_13_35_00.out
Server Protection_NMC server backup_2018_01_12_14_00_00__192048.log.gz
Server Protection_NMC server backup_2018_01_12_14_00_00_flt_msg.txt
Server Protection_NMC server backup_2018_01_12_14_00_00.out
Server Protection_Server backup_2018_01_12_10_00_00__192001.log.gz
Server Protection_Server backup_2018_01_12_10_00_00_flt_msg.txt
Server Protection_Server backup_2018_01_12_10_00_00.out
```

Per Default werden die Protokolle der letzten 30 Tage aufgehoben. Natürlich läßt sich auch diese Zeitspanne durch einen Konfigurationsparameter beliebig anpassen (siehe **WKF_LOG_RETENTION_IN_DAYS**).

II.6.1.2. Beschreibung der Protokolldateien und deren Inhalte

- **Die NetWorker Notification Ausgabe**

  Jede Notification Ausgabe wird nach folgendem Namensschema abgelegt:

  <Policy-Name>_<Workflow-Name>_<Zeitstempel des Workflow-Start>**.out**

  Beispiel:

  Server Protection_Server backup_2018_01_11_11_48_56.out

- **Die Zusammenfassung der Workflow-Meldungen**

  Das **nsr_workflow_analayse** Programm ermittelt anhand des erhaltenen Workflow-Namens und den gegebenen Zeiten, alle zu diesem Workflow gehörigen Protokolldateien und fasst diese in einer Datei zusammen. Der Name der Meldungsdatei ergibt sich hierbei nach folgendem Namensschema:

<Policy-Name>_<Workflow-Name>_<Zeitstempel des Workflow-Start>__<Job-Id>**.log.gz**

Da diese Dateien recht groß werden können, werden sie nach der Bearbeitung mit dem gzip Kompressionsverfahren komprimiert.

Beispiel:

Server Protection_Server backup_2018_01_11_11_48_56__352041.log.gz

- **Das Ergebnis der Meldungs-Kompilierung**

Die dritte Protokolldatei ist das Ergebnis des Kompiliervorgangs. In dieser Datei sind die erkannten Probleme und Fehler, sowie die erfolgreichen SaveSet Informationen enthalten. Die letzte Zeile der Datei enthält den Exit-Wert, der sich aus der Compilation ergebenen hat.

Beispiel:

```
Completion Message Info for Protection_Group: 'FS_Backup_Clone'
* 01/11/18 13:36:07  nsrworkflow Action 'FS/Backup_Clone/backup' failed.
* 01/11/18 13:37:02  nsrworkflow Workflow 'FS/Backup_Clone' failed.
* 01/11/18 13:35:03  savegrp ubuntu-14:All Unable to create session channel wi
* 01/11/18 13:35:03  savegrp Unable to determine status for job 160042, ubuntu
* 01/11/18 13:35:03  savegrp Job 160042 with command 'savefs -s nsr9-121.schae
* 01/11/18 13:35:03  savegrp Job 160042 host: ubuntu-14 savepoint: ubuntu-14:A
* 01/11/18 13:35:03  savegrp ubuntu-14:All unexpectedly exited.
* 01/11/18 13:35:03  savegrp ubuntu-14:All will retry 1 more time(s).
* 01/11/18 13:35:03  savegrp ubuntu-14:All next retry in 1 seconds.
* 01/11/18 13:35:06  savegrp ubuntu-14:All Unable to create session channel wi
.
.
.
  nsr9-121.schaefer-tobies.lan: /   level=incr,    655 MB 00:00:44    390 file
  completed savetime=1515674105
  nsr9-121.schaefer-tobies.lan: /boot/grub2/i386-pc level=incr, 500 KB 00:00:0
  completed savetime=1515674114
.
.
.
MSG_Compiler Exit Code: 3
```

Der Name des Kompilierergebnisses ergibt sich hierbei aus folgendem Namensschema:

<Policy-Name>_<Workflow-Name>_<Zeitstempel des Workflow-Start>**_flt_msg.txt**

Beispiel:

Server Protection_Server backup_2018_01_11_11_48_56_flt_msg.txt

Der Inhalt dieser Datei wird als Eingabe an die Informations-Kommandos weitergeleitet. D.h. der Dateiinhalt entspricht dem Inhalt der Mail bzw. der Eingabe des LOGGER-Kommandos.

II.6.2. Debugging

II.6.2.1. Debugdatei des Programm-Laufes

Das Programm **nsr_workflow_analyse** schreibt Debug-Informationen eines Programm-Laufes in die Datei */tmp/schtob/nsr_workflow_analyse.dbg*. Der Pafd des Ablageverzeichnis kann durch die Definition des Parameters **DBG_DIR** angepasst werden.

Bei jedem Start des Kommandos wird zunächst die Größe der bestehenden Debug-Datei kontrolliert. Sollte diese größer als 5 MB sein, so wird die bestehende Debug-Datei komprimiert und auf den Namen nsr_workflow_analyse.1.gz umbenannt. Ingesammt werden bis zu 3 ältere Debug-Dateien rollierend aufbewahrt. Ist die Dateigröße noch nicht erreicht, so wird die Debug-Ausgabe an die bestehende Ausgabe angefügt.

Interessante Meldungen in dieser Debug-Datei können z.B. nicht erkannte Meldungen in den Workflow-Ausgaben sein. Diese Meldungen werden durch den Text **"__find_next_matching_regexp__: No match found for line"** eingeleitet. Die nicht erkannte Meldung findet man dann in der folgenden Zeile.

Beispiel:

```
02/12/18 13:51:41  __find_next_matching_regexp__: No match found for line:
    <* 02/12/18 13:35:22  savegrp nsr9-12:/ abandoned.
>
```

Am Ende der Debug-Ausgabe eines Programmlaufs erhält man außerdem Informationen über das gestartete Informations-Kommando.

Beispiel:

```
02/12/18 14:00:30  send_mail: cmd </usr/bin/mailx -a "/nsr/logs/workflow_logs/2
```

II.6.2.2. Debugging der nsr_workflow_analyse Programms

Sollte es Probleme mit dem **nsr_workflow_analyse** Programm oder mit dem Inhalt der Filter-Datei geben, so können sie das Programm von Hand mit einer Debug-Option starten. Hierdurch wird zum Beispiel das Einlesen aller regulären Ausdrücke aus der Filter-Datei protokolliert. Als Eingabe für einen solchen Debug-Aufruf muss die Notification-Ausgabedatei verwendet werden (siehe **Notification Ausgabe**).

Beispiel:

```
# cat FS_Backup_2018_02_12_13_35_00.out | nsr_workflow_analyse.py --debug 2
```

II.6.2.3. Debugging des Compiler-Ergebnisses

Wenn eine Ausgabe-Zeile nicht mit dem erwarteten Exit-Wert bewertet wird, oder ein mehrzeiliger regulärer Ausdruck nicht zum Zuge kommt, so kann man den Compilation-Vorgang zusätzlich debuggen. Hierzu kann das Compiler-Modul **msg_compile.py** mit folgenden Argumenten aufgerufen werden.

```
Usage:
    msg_compile.py [-v]* [(-f | --filter) <filter_file>]
            [(-i | --input) <input_file>]
            [(-o | --output) <output_file>]
            [(-g | --protection_group) <group_name>]
            [--debug [1-6]]

-f <filter_file>    Filename which includes the REGEXP und Profil Defs
-i <input_file>     Combined Workflow Messages file
                                (generated by nsr_workflow_analyse.py)
```

```
                    Default: STDIN
-o <output_file>    Filename to store the Result of the Filter-Action
                    Default: STDOUT
-g <group name>     Protection Group Name (Only needed for Debugging)
```

Das Python-Modul findet man im Installationsverzeichnis im Unterverzeichnis schtob/nsr_tools. Der Aufruf eines von Hand gestarteten Compiler-Lauf erwartet als Eingabe, die zusammengfasste Ausgabe eines Workflow-Laufes. Ein beispielhafter Aufruf würde demzufolge in etwa so aussehen:

Beispiel:

```
# python .../msg_compile.py -i FS_Backup_..._416043.log.gz --debug 3
```

Die Debug-Werte ergeben folgende zusätzliche Ausgaben:

- **--debug 1**

  Ausgabe des Exit-Wertes vor jeder erkannten Fehlerzeile oder Zeilengruppe.

  Beispiel:

```
3* 12.02.2018 13:35:22  savegrp Job 416053 host: w12r2-itk savepoint: w12r2-it
 * 12.02.2018 13:35:22  savegrp w12r2-itk:All unexpectedly exited
1* 12.02.2018 13:50:06  savegrp Job 416069 host: nsr9-121 savepoint: /var/log
3* 12.02.2018 13:50:06  savegrp nsr9-121 failed.
2* 12.02.2018 13:50:06  savegrp nsr9-121 will retry 1 more time(s).
2* 12.02.2018 13:50:06  savegrp nsr9-121 next retry in 1 seconds.
1* 12.02.2018 13:50:13  savegrp Client 'nsr9-12' is being skipped because its
```

  Die ersten beiden obigen Zeilen ergeben zusammen den Exit-Wert 3. Die folgenden Zeilen werden von einzeiligen regulären Ausdrücken erfasst und ergeben jeweils den vorangestelleten Exit-Wert.

- **--debug 2**

  Gibt für jede Zeile der Workflow-Ausgabe den zugehörigen Exit-Wert und die Zeilen-Nummer des gefundenen regulären Ausdrucks in der Filter-Datei aus.

  Beispiel:

```
0: 51: * 12.02.2018 13:51:34  nsrworkflow Action 'FS/Backup_Clone/clone' succe
3: 66: * 12.02.2018 13:51:34  nsrworkflow Workflow 'FS/Backup_Clone' failed
```

  Die erste Zeile wurde anhand des regulären Ausdrucks in der Zeile 51 der Filter-Datei als fehlerfrei erkannt. Die darauffolgende Zeile wurde durch den regulären Ausdruck in Zeile 66 der Filter-Datei als Fehler mit dem Exit-Wert 3 versehen.

- **--debug 3**

  Zusätzliche Debug-Ausgabe im zugehörigen Debug-File */tmp/schtob/msg_compile.dbg*. In diesem Falle wird die Defintion der Workflow-Profile zusätzlich ausgegeben.

- **--debug 4**

  Zusätzliche Debug-Ausgabe im zugehörigen Debug-File */tmp/schtob/msg_compile.dbg*. In diesem Falle wird das Debugging für mehrzeilige reguläre Ausdrücke erhöht. Man erhält Informationen zu

Ausgaben, deren erste bis 'n'te Zeile zu einer mehrzeiligen Regel passten. Deren 'm'te Zeile dann aber nicht mehr zutraf.

- **--debug 5**

  Zusätzliche Debug-Ausgabe im zugehörigen Debug-File */tmp/schtob/msg_compile.dbg*. Weitere zusätzliche Debug-Ausgaben.

Kapitel III. Das start_workflow Kommando

V1.1.1

III.1. Beschreibung der Vorteile

Das **nsr_workflow_analyse** Programm stellt eine komfortable Schnittstelle zum Start von NetWorker Workflows dar. Gegenüber dem NetWorker eigenen nsr_workflow Kommando besitzt es folgende Vorteile:

- Das Kommando kann auf jedem beliebigen Rechner (auch nicht NetWorker Client) gestartet werden. Es muss lediglich ein NetWorker Administrator-Benutzer mitgegeben oder konfiguriert werden.
- Die zu startende Policy, der Workflow und die zu startenden Client-Namen können interaktiv ausgewählt werden (s.u.)
- Der Workflow-Probe Aufruf gibt das Ergebnis des Laufes direkt an der Console aus.
- Das Kommando gibt die aktiven Workflow-Prozesse alle "n"-Sekunden an der Console aus.
- Wenn auf einer NetWorker Client-Maschine mehrere Datenbanken laufen, kann der Start einer einzelnen Datenbank einfach über die Datenbank-ID erfolgen.

III.2. Voraussetzung

Die Konfigurationsdatei **nsr_api_local_defines.py** wurde angelegt.

III.3. Usage

```
# start_workflow.py -?

    Usage:
        start_workflow.py -i [-v] [--wait]
        start_workflow.py [-a] [-v] [ <other Args> ]
        start_workflow.py -s <nsr-srv> [-v]* [(--user | -u) <user>]
                          [(--password | -P) <password>] [ <other Args> ]

    <other Args> :==
        (--policy | -p) <policy-name>
        (--workflow | -w) <workflow>
        [--action_name <action-name>]
        [(--client | -c) <client-name>]*
        [(--level | -l) <level> ]
        [--probe]
        [(--retention | -y) <ret-time>]
        [--wait]
        [--db_id <DB-ID>]
        [-n]

    --user | -u       NetWorker Administrator user name
    --password | -P   NetWorker User Password
    -i                interactive; Ask for all parameters
    -a                active;
                      Parameters are defined in the 'nsr_api_local_defines.py'
                      file
    --policy | -p     backup policy name
```

```
    --workflow | -w   workflow name
    --action_name     give the Action-name; sometimes needed for -l and -p star
    --client | -c     start only this client from workflow-group

    --level | -l      start backup with Backup-Level
                      (requires: Action-Name: 'backup' or given Action-name)
    --probe           start backup action with Probe-Option
    -v | --wait       verbose; Wait until the job finishes!
                      if more then one '-v' is given the Action is called
                      with the verbose option too.
    --retention | -y  override the retention time

    --db_id <DB-ID>   Start DB-Backup for the given client
    -n                No save; (see man savegrp)
```

Beispiele:

Interaktiver Aufruf des Programms. Hierbei werden alle Parameter abgefragt.

```
# start_workflow.py -i

    Please enter NetWorker Server name
    nsr name: nsr9-121

    Please insert administrator user name
    user: administrator
Password:

Please enter Policy-Name or Nr. of the Policy listet:
    0) Bronze
    1) Clone
    2) DB
    3) FS
    4) Gold
    5) Platinum
    6) Server Protection
    7) Silver
    8) qSkills
    9) test
Policy name: 0

Please enter Workflow-Name or Nr. of the Workflow listet:
    0) Applications
    1) Filesystem
Workflow name: 1

Please enter the Nr(s). of the listet Clients yout want to start:
    If you want to start more then One client enter them comma separated

    0) All Clients
    1) nsr9-12                      : /
    2) nsr9-121.schaefer-tobies.lan : All
    3) open-suse                    : All
    4) ubuntu-14                    : All
    5) w12r2-itk                    : All
    6) win10-a                      : All
client list: 1,2
```

```
Job Id: 416128
```

Aktiver Aufruf des Programms. Probe eines Clients (Auflösung des SaveSets All in seine zu sichernden SaveSets).

```
# start_workflow.py --probe -p FS -w Backup_Clone -c nsr9-121.schaefer-tobies.la

------------------------------------------------------------------------------
Nr. of workflow jobs: 3
State of jobs

    name    : nsr9-121.schaefer-tobies.lan:All
    command : savefs -s nsr9-121.schaefer-tobies.lan -c nsr9-121.schaefer-tobies
    state   : Queued


    name    : savegrp
    command : savegrp -Z backup:traditional -v -p
    state   : Active


    name    : FS
    command : /usr/sbin/nsrworkflow -p FS -w Backup_Clone -a -L -c nsr9-121.scha
    state   : Active


------------------------------------------------------------------------------
Nr. of workflow jobs: 3
State of jobs

    name    : savegrp
    command : savegrp -Z backup:traditional -v -p
    state   : Active


    name    : FS
    command : /usr/sbin/nsrworkflow -p FS -w Backup_Clone -a -L -c nsr9-121.scha
    state   : Active


------------------------------------------------------------------------------
Nr. of workflow jobs: 4
State of jobs

    name    : nsrclone
    command : nsrclone -a "*policy name=FS" -a "*policy workflow name=Backup_Cl
    state   : Queued


    name    : FS
    command : /usr/sbin/nsrworkflow -p FS -w Backup_Clone -a -L -c nsr9-121.scha
    state   : Active


------------------------------------------------------------------------------
Nr. of workflow jobs: 5
State of jobs
```

```
    name    : nsrclone
    command : nsrclone -a "*policy name=FS" -a "*policy workflow name=Backup_Cl
    state   : Active


    name    : FS
    command : /usr/sbin/nsrworkflow -p FS -w Backup_Clone -a -L -c nsr9-121.scha
    state   : Active


------------------------------------------------------------------------------
Nr. of workflow jobs: 5
State of jobs

    name    : FS
    command : /usr/sbin/nsrworkflow -p FS -w Backup_Clone -a -L -c nsr9-121.scha
    state   : Active


------------------------------------------------------------------------------
Nr. of workflow jobs: 5
State of jobs

==============================================================================
                            Completion Info
==============================================================================

    name    : nsrclone
    command : NO COMMAND given
    status  : Completed
    name    : nsrclone
    command : nsrclone -a "*policy name=FS" -a "*policy workflow name=Backup_Cl
    status  : Unknown
    name    : nsr9-121.schaefer-tobies.lan:All
    command : savefs -s nsr9-121.schaefer-tobies.lan -c nsr9-121.schaefer-tobie
    status  : Succeeded
    name    : nsr9-121.schaefer-tobies.lan:All
    command : savegrp -Z backup:traditional -v -p
    status  : Succeeded
    message : 12.02.2018 17:00:14  savegrp --- Probe Summary ---\n
 12.02.2018 17:00:14  savegrp nsr9-121.schaefer-tobies.lan:All
 12.02.2018 17:00:14  savegrp level=full, vers=ssbrowse, p=12
 12.02.2018 17:00:14  savegrp nsr9-121.schaefer-tobies.lan:/
 12.02.2018 17:00:14  savegrp level=incr, vers=ssbrowse, p=12
 12.02.2018 17:00:14  savegrp nsr9-121.schaefer-tobies.lan:/   level=incr, pool=
 12.02.2018 17:00:14  savegrp nsr9-121.schaefer-tobies.lan:/home
 12.02.2018 17:00:14  savegrp level=incr, vers=ssbrowse, p=12
 12.02.2018 17:00:14  savegrp nsr9-121.schaefer-tobies.lan:/home level=incr, poo
 12.02.2018 17:00:14  savegrp nsr9-121.schaefer-tobies.lan:/opt
 12.02.2018 17:00:14  savegrp level=incr, vers=ssbrowse, p=12
 12.02.2018 17:00:14  savegrp nsr9-121.schaefer-tobies.lan:/opt level=incr, poo
 .
 .
 .
 12.02.2018 17:00:14  savegrp Client 'nsr9-121.schaefer-tobies.lan' is being ski
 12.02.2018 17:00:19  savegrp Action backup traditional 'backup' with job id 410
```

```
    name    : FS
    command : /usr/sbin/nsrworkflow -p FS -w Backup_Clone -a -L -c nsr9-121.scha
    status  : Succeeded
```

Start einer Full-Sicherung eines definierten Clients ohne Angabe eines Workflows. Der Job wartet durch die Angabe der Option --wait bis der workflow beendet wurde. In der Zwischenzeit gibt es alle 5 Sekunden den aktuellen Status des Workflow-Laufs aus.

```
# start_workflow.py --wait -l full -c open-suse

Please enter Policy-Name or Nr. of the Policy listet:
    0) Bronze
    1) Clone
    2) DB
    3) FS
    4) Gold
    5) Platinum
    6) Server Protection
    7) Silver
    8) qSkills
    9) test
Policy name: 3

Please enter Workflow-Name or Nr. of the Workflow listet:
    0) Backup_Clone
Workflow name: 0
```

```
Nr. of workflow jobs: 3
State of jobs

    name    : open-suse:All
    command : savefs -s nsr9-121.schaefer-tobies.lan -c open-suse -g FS_Backup_
    state   : Queued


    name    : savegrp
    command : savegrp -Z backup:traditional -v -l full
    state   : Active


    name    : FS
    command : /usr/sbin/nsrworkflow -p FS -w Backup_Clone -a -L -c open-suse -A
    state   : Active
```

```
Nr. of workflow jobs: 25
State of jobs

    name    : /opt
    command : save -LL -s nsr9-121.schaefer-tobies.lan -g FS/Backup_Clone/backu
    state   : Queued


    name    : /home
    command : save -LL -s nsr9-121.schaefer-tobies.lan -g FS/Backup_Clone/backu
    state   : Queued
```

```
.
.
.

    name    : /
    command : save -LL -s nsr9-121.schaefer-tobies.lan -g FS/Backup_Clone/backu
    state   : Active


    name    : savegrp
    command : savegrp -Z backup:traditional -v -l full
    state   : Active


    name    : FS
    command : /usr/sbin/nsrworkflow -p FS -w Backup_Clone -a -L -c open-suse -A
    state   : Active

.
.
.
```

```
Nr. of workflow jobs: 25
State of jobs

    name    : /
    command : save -LL -s nsr9-121.schaefer-tobies.lan -g FS/Backup_Clone/backu
    state   : SessionActive


    name    : savegrp
    command : savegrp -Z backup:traditional -v -l full
    state   : Active


    name    : FS
    command : /usr/sbin/nsrworkflow -p FS -w Backup_Clone -a -L -c open-suse -A
    state   : Active

.
.
.
```

```
Nr. of workflow jobs: 26
State of jobs

    name    : nsrclone
    command : nsrclone -a "*policy name=FS" -a "*policy workflow name=Backup_Cl
    state   : Queued


    name    : FS
    command : /usr/sbin/nsrworkflow -p FS -w Backup_Clone -a -L -c open-suse -A
    state   : Active
```

```
Nr. of workflow jobs: 27
State of jobs

    name    : nsrclone
    command : nsrclone -a "*policy name=FS" -a "*policy workflow name=Backup_Cl
    state   : Active


    name    : FS
    command : /usr/sbin/nsrworkflow -p FS -w Backup_Clone -a -L -c open-suse -A
    state   : Active

.
.
.
================================================================================
                            Completion Info
================================================================================

    name    : nsrclone
    command : NO COMMAND given
    status  : Completed
    name    : nsrclone
    command : nsrclone -a "*policy name=FS" -a "*policy workflow name=Backup_Cl
    status  : Succeeded
.
.
.
    name    : open-suse:/srv
    command : save -LL -s nsr9-121.schaefer-tobies.lan -g FS/Backup_Clone/backu
    status  : Succeeded
    name    : open-suse:/opt
    command : save -LL -s nsr9-121.schaefer-tobies.lan -g FS/Backup_Clone/backu
    status  : Succeeded
    name    : open-suse:/home
    command : save -LL -s nsr9-121.schaefer-tobies.lan -g FS/Backup_Clone/backu
    status  : Succeeded
    name    : open-suse:/
    command : save -LL -s nsr9-121.schaefer-tobies.lan -g FS/Backup_Clone/backu
    status  : Succeeded
    name    : open-suse:All
    command : savefs -s nsr9-121.schaefer-tobies.lan -c open-suse -g FS_Backup_
    status  : Succeeded
    name    : savegrp
    command : savegrp -Z backup:traditional -v -l full
    status  : Succeeded
    name    : FS
    command : /usr/sbin/nsrworkflow -p FS -w Backup_Clone -a -L -c open-suse -A
    status  : Succeeded
```

III.4. Protokolle und Debugging

Jeder Lauf des Programms schreibt seine Aufrufparameter und das Ergebnis des Workflow-Laufes in die Debugging Datei **/tmp/schtob/start_workflow.dbg**. Der Pfad des Ablageverzeichnis kann durch die Definition des Parameters **DBG_DIR** angepasst werden.

Durch die Angabe von einem oder mehreren "-v" Optionen können die Debugging Ausgaben des Programms erhöht werden.

Bei jedem Start des Kommandos wird zunächst die Größe der bestehenden Debug-Datei kontrolliert. Sollte diese größer als 5 MB sein, so wird die bestehende Debug-Datei komprimiert und auf den Namen start_workflow.1.gz umbenannt. Ingesammt werden bis zu 3 ältere Debug-Dateien rollierend aufbewahrt. Ist die Dateigröße noch nicht erreicht, so wird die Debug-Ausgabe an die bestehende Ausgabe angefügt.

Kapitel IV. Die Schäfer &Tobies NetWorker REST-API Library

V1.1.1

IV.1. Die REST-API Library

IV.1.1. Python "schtob" Library

Sie können die Schäfer & Tobies NetWorker REST-API auch für eigene Projekte verwenden. Für Fragen rund um die Library senden sie am besten eine Mail an ***info@schaefer-tobies.de*** , oder an den Autor dieses Dokumentes.