

# Administration Manual

## eRunbook Version 6.0

Stand: 01.07.09

Dokument Version: 513





## Inhaltsverzeichnis

- 1 Vorwort.....1
- 2 Einleitung.....2
- 3 eRunbook Betrieb.....3
  - 3.1 Server Komponenten.....3
  - 3.2 Netzwerk Kommunikation.....4
  - 3.3 Schnittstellen.....4
  - 3.4 Start und Stop Routinen.....6
  - 3.5 Sicherungsmaßnahmen.....6
- 4 eRunbook Server.....7
  - 4.1 Command Line.....7
  - 4.2 Prozesse .....7
  - 4.3 Environment und Settings.....8
    - 4.3.1 \$NOVAHOME.....8
    - 4.3.2 \$NOVAINSTANZ.....8
    - 4.3.3 \$LD\_LIBRARY\_PATH.....8
    - 4.3.4 IP Adressen in etc/hosts.....9
  - 4.4 Backup und Restore.....10
  - 4.5 eRunbook LDAP Import Interface.....11
    - 4.5.1 Funktionsweise.....11
    - 4.5.2 eRunbook\_ldap\_get\_user.pl.....13
  - 4.6 XML Import Interface.....14
  - 4.7 Verzeichnisstruktur unter \$NOVAHOME.....17
    - 4.7.1 \$NOVAHOME/instanz/example.....18
    - 4.7.2 \$NOVAHOME/product.....19
  - 4.8 Löschen einer eRunbook Instanz.....20
  - 4.9 Existierende Klasse umbenennen.....21
  - 4.10 Existierende Beschreibung (Description) einer Instanz ändern.....21
  - 4.11 Löschen des gesamten Inhaltes der Klasse “trash”.....21
  - 4.12 Löschen der eRunbook Server Software.....22
  - 4.13 Drucken von eRunbook Dokumenten.....23
  - 4.14 Scan Funktionalität.....27
    - 4.14.1 Online Agent Scan.....27
    - 4.14.2 Geplante nächtliche Online Scans aller Rechner.....27
    - 4.14.3 Offline Scan.....27
  - 4.15 Apache Webserver .....29
    - 4.15.1 Start/Stop.....29
    - 4.15.2 Benutzerauthentifizierung über Apache mit “htpasswd”.....29
    - 4.15.3 Änderung des SSL Zertifikates.....30
  - 4.16 mySQL Datenbanksystem.....31
    - 4.16.1 Start/Stop.....31



4.16.2 Backup.....	31
4.16.3 Standard Tabellengrößenbegrenzung ändern.....	31
4.17 Oracle Datenbanksystem.....	32
4.17.1 Kompatibilität Oracle Client zu Oracle Server Versionen.....	32
4.17.2 Voraussetzungen für den Betrieb.....	32
4.17.3 Zugriffsrechte auf die Oracle Datenbank.....	32
4.17.4 Oracle Tablespaces.....	33
4.17.5 eRunbook.conf Einstellungen für die Oracle Anbindung.....	34
4.17.6 Anleitung zur Integration der Oracle Datenbank.....	35
4.18 Datenbank allgemein.....	36
4.18.1 db_thin_out.pl.....	36
5 Anhang.....	38
5.1 Hardware Sizing.....	38
5.2 Datenbankgröße.....	38
5.3 Schreibperformance.....	39
5.4 Leseperformance.....	41
5.4.1 Leseperformance der Suche.....	41
5.4.2 Leseperformance des DocViewers.....	42



## 1 Vorwort

eRunbook ist ein eingetragenes Markenzeichen der nova ratio AG.

Die Nutzung der eRunbook Software und aller sich darauf beziehenden Dokumente ist durch eine eRunbook Lizenzvereinbarung reglementiert.

nova ratio AG  
Haus Petersberg  
Hauptstraße 3-5  
56235 Hundsdorf  
Germany

Telefon: +49 2623 9242 0  
Telefax: +49 2623 9242 100  
eMail: [info@nova-ratio.de](mailto:info@nova-ratio.de)  
Support Call: [support@nova-ratio.de](mailto:support@nova-ratio.de)

## 2 Einleitung

Das Dokument *eRunbook Administration Manual* beschreibt die Administration einer eRunbook Umgebung und erklärt alle Vorgehensweisen, die für einen Betrieb relevant sind.

**Hinweis:** Dieses Dokument verwendet in seinen Beispielen eine eRunbook Instanz mit dem Namen `example`. Dieser Name ist nur als Platzhalter für den gewollten Instanznamen zu verstehen.

Das *eRunbook Administration Manual* dient darüber hinaus der Fehleranalyse und Fehlereingrenzung. Es liefert jedoch keine Funktions- und Vorgehensweisen, wie die Funktionen des eRunbook Systems erweitert oder geändert werden können.

Alle notwendigen Komponenten werden während einer Neuinstallation eingerichtet. Dies umfasst sowohl automatische Start und Stop-Prozeduren, als auch den für den Betrieb erforderlichen Webserver und die Datenbankapplikation. Für eine umfassende Beschreibung der Installation verwenden Sie bitte das *eRunbook Installation Manual*.

## 3 eRunbook Betrieb

**Hinweis:** Für einen erfolgreichen Betrieb des eRunbook Systems muss die Einhaltung der Angaben in den nachfolgenden Kapiteln sichergestellt werden.

### 3.1 Server Komponenten

Wenn man die Aufgabe hat eRunbook zu betreiben, sind folgende Komponenten für den Grundbetrieb erforderlich:

1. Der eRunbook Webserver (lokal)  
Er stellt alle Informationen über das Webfrontend zur Verfügung.  
*Weitere Informationen finden Sie Kapitel 4.15 dieser Dokumentation*
2. Die eRunbook Datenbank (lokal)  
Sie dient als zentraler Ort zur Speicherung der Daten und bildet die Grundlage der Historie.  
*Weitere Informationen finden Sie im Kapitel 4.16 dieser Dokumentation*
3. Die eRunbook Server Prozesse  
Die Server Prozesse haben den User „erunbook“ als Eigentümer.  
*Weitere Informationen finden Sie im Kapitel 3.2 dieser Dokumentation*

Die oben genannten Komponenten werden auf dem eRunbook Server im Rahmen der Installationsroutine automatisch installiert und eingerichtet.

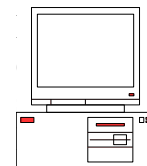
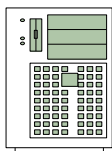
**Hinweis:** Bei der Installation werden automatisch Start- und Stop-Skripte für die eRunbook Server Komponenten erstellt. Ein separater Start ist nach dem Boot-Vorgang nicht mehr erforderlich.

### 3.2 Netzwerk Kommunikation

Der eRunbook Server kommuniziert mit den Zielsystemen auf einer während der Installation festgelegten Anzahl von Ports. Als Standard sind folgende Ports vorgesehen:

1. Agent Port (eingehend): 9001
2. Server Port (eingehend): 9002
3. Webserver Secure Port: 9443

Die Kommunikation des Servers mit den Agenten verläuft in nachstehend dargestellter Reihenfolge:



- (1) Server sendet Scan-Auftrag
- (4) Server empfängt auf Port 9002

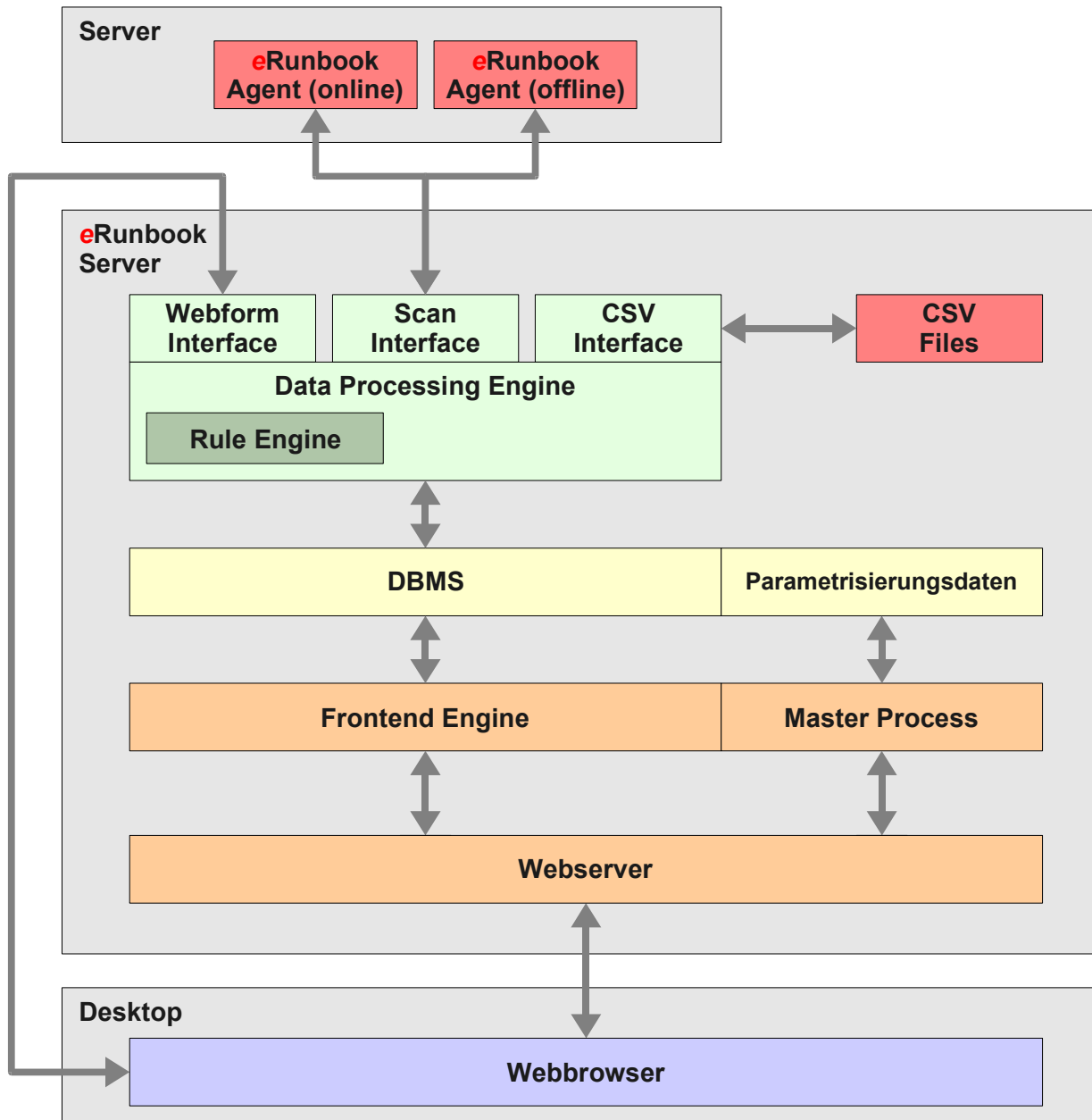
- (2) Agent empfängt auf Port 9001
- (3) Agent sendet Antwort

### 3.3 Schnittstellen

Über folgende Schnittstellen werden Informationen in das System eingelesen:

1. **eRunbook Agenten auf den Zielsystemen**  
Zur Abfrage von Konfigurationsdaten ist die Installation eines Agenten auf dem Zielsystem erforderlich. Über die Weboberfläche von eRunbook wird mittels einer Ampeldarstellung die Erreichbarkeit des Online Agenten auf dem Zielsystem angezeigt.  
*Weitere Informationen finden Sie im InstallationManual, Kapitel 3.1.2*
2. **CSV-Import**  
Ein User mit entsprechender Berechtigung hat die Möglichkeit, dem System per Import Daten zur Verfügung zu stellen.  
*Weitere Informationen finden Sie im UserManual, Kapitel 5.3.1*

Die Abbildung der folgenden Seite skizziert das Zusammenspiel der installierten Komponenten.



### 3.4 Start und Stop Routinen

Der Systemadministrator hat die Möglichkeit das eRunbook System inkl. aller Komponenten über die Kommandozeile zu starten oder zu stoppen.

*Weitere Informationen finden Sie im Kapitel 4.1 dieser Dokumentation*

### 3.5 Sicherungsmaßnahmen

Zur Sicherung eines eRunbook Systems müssen Verzeichnisstrukturen und Datenbanktabellen des eRunbook Servers gesichert werden. Die Datenbank der runtime Umgebung kann über einen MySQL Dump gesichert werden.

*Weitere Informationen finden Sie im Kapitel 4.4 dieser Dokumentation*

## 4 eRunbook Server

### 4.1 Command Line

Das eRunbook System wird als Benutzer “root” mit dem Script “/etc/init.d/eRunbook start” gestartet und mit “/etc/init.d/eRunbook stop” gestoppt.

Mit diesem Befehl werden alle benötigten Prozesse respektive Komponenten des eRunbook Systems gestartet bzw. gestoppt.

```
ko-pc-10:/etc/init.d # ./eRunbook start
Starting service eRunbook
Starting service eRunbookAgent
Starting service MySQL
eRunbook daemons: instance eAdmin          ... started
eRunbook daemons: instance Test           ... started
eRunbook communication daemon             ... started
Starting service OpenOffice

ko-pc-10:/etc/init.d # ./eRunbook stop
Stopping service eRunbook
Stopping service OpenOffice
eRunbook daemons: instance eAdmin          ... stopped
eRunbook daemons: instance Test           ... stopped
eRunbook communication daemon             ... stopped
Shutting down service mysql
Stopping service eRunbookAgent
```

### 4.2 Prozesse

Ist der Server gestartet, so sind folgende Prozesse auf dem System aktiv:

Prozess	Verwendung
/opt/eRunbook/software/additional/mysqlserver/bin/	MySQL Server
/opt/eRunbook/product/application/eRunbook.server [Instanzname]	Perl Schnittstelle
/opt/eRunbook/product/application/eRunbook.cron [Instanzname]	Perl Schnittstelle
/opt/eRunbook/product/communication/cserver	Communication Serv.
/opt/eRunbook/software/additional/httpd/bin/httpd	Webserver

**Hinweis:** Je nach Umfang der vorhandenen eRunbook Installation können für mehrere Instanzen auch mehrere Prozesse aktiv sein.

## 4.3 Environment und Settings

Das eRunbook System benötigt auf der Kommandozeilenebene einige Einstellungen. Die folgenden Beispiele wurden mit einer C-Shell als Login Shell erstellt.

### 4.3.1 \$NOVAHOME

Die Environmentvariable \$NOVAHOME beinhaltet den Pfad zum eRunbook Installationsverzeichnis, welches die gesamte eRunbook Software beinhaltet.

```
setenv NOVAHOME "/opt/eRunbook"
```

### 4.3.2 \$NOVAINSTANZ

Die Variable \$NOVAINSTANZ enthält den Pfad zu dem Verzeichnis, in dem die Instanzdaten gespeichert sind.

```
setenv NOVAINSTANZ "/opt/eRunbook/instanz"
```

### 4.3.3 \$LD\_LIBRARY\_PATH

Die Variable \$LD\_LIBRARY\_PATH enthält den Pfad zu den Verzeichnissen, welche die Libraries beinhalten, die nicht in den default Library Pfaden enthalten sind. Um mit eRunbook zu arbeiten muss diese Umgebungsvariable um die Pfade der eRunbook Libraries erweitert werden.

```
setenv LD_LIBRARY_PATH "/opt/eRunbook/software/mandatory/openssl/lib:  
/opt/eRunbook/software/mandatory/oracleClient/jdbc/lib:  
/opt/eRunbook/software/mandatory/oracleClient/lib:  
/opt/eRunbook/software/mandatory/freetype/lib:  
/opt/eRunbook/software/mandatory/mysqlClient/lib:  
/opt/eRunbook/software/mandatory/graphviz/lib:  
/opt/eRunbook/software/mandatory/other/lib:  
/opt/eRunbook/software/mandatory/perl/lib:  
/opt/eRunbook/software/mandatory/java/lib:  
/opt/eRunbook/software/mandatory/gd/lib:  
/opt/eRunbook/software/additional/mysqlServer/lib:  
/opt/eRunbook/software/additional/httpd/lib:$LD_LIBRARY_PATH"
```



### 4.3.4 IP Adressen in etc/hosts

Da einige Programmaufrufe die erste gültige IPv4 Adresse des Systems für Funktionen auslesen (ausgenommen 127.0.0.1), sollte die Reihenfolge in der hosts Datei folgendermaßen aussehen:

```
127.0.0.1      localhost

# special IPv6 addresses
::1           localhost ipv6-localhost ipv6-loopback

fe00::0       ipv6-localnet

ff00::0       ipv6-mcastprefix
ff02::1       ipv6-allnodes
ff02::2       ipv6-allrouters
ff02::3       ipv6-allhosts
10.0.2.14     Beispielrechnername #1      <---- Adresse des Systems im Netzwerk
127.0.0.2     Domain.Beispielrechnername #2
```

### 4.4 Backup und Restore

Um eine angepaßte Backupstrategie für eRunbook zu entwickeln, kann die Software in die folgenden Komponenten unterteilt werden:

- Dateien im Verzeichnis \$NOVAHOME und dessen Unterverzeichnisse
- Dateien im Verzeichnis \$NOVAINSTANZ und dessen Unterverzeichnisse
- Datenbank der Instanz im entsprechenden Datenbankmanagementsystem, welches von eRunbook benutzt wird.

**HINWEIS:** Falls die default Parameter der eRunbook Installation beibehalten wurden ist zu berücksichtigen, daß sowohl das \$NOVAINSTANZ Verzeichnis als auch die MySQL Datenbank unterhalb des Verzeichnisses \$NOVAHOME liegen.

Die Dateien im Verzeichnis \$NOVAHOME/product und \$NOVAHOME/software können mit Hilfe einer Neuinstallation der eRunbook Software wiederhergestellt werden. Daher muß dieses Verzeichnis nicht unbedingt in die Datensicherungsstrategie integriert werden.

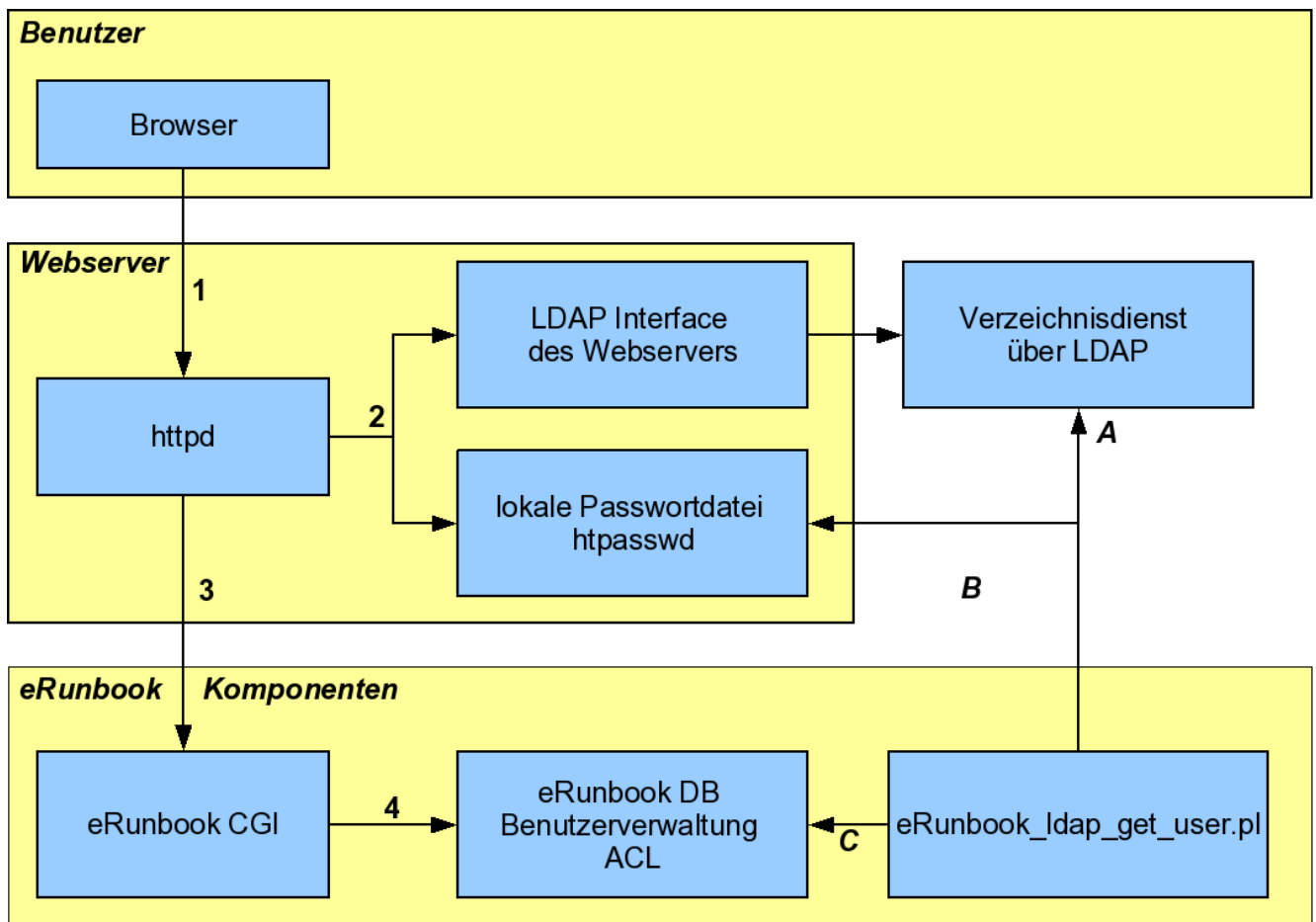
Die Daten im Verzeichnis \$NOVAINSTANZ können NICHT durch eine Neuinstallation der eRunbook Software wiederhergestellt werden. Dieses Verzeichnis muß also durch entsprechende Maßnahmen in eine Backupstrategie aufgenommen werden.

Für die Sicherung der Datenbankinformationen jeder eRunbook Instanz stehen die Backup Tools der entsprechenden DBMS zur Verfügung.

## 4.5 eRunbook LDAP Import Interface

### 4.5.1 Funktionsweise

Um die vorhandenen Benutzer eines Verzeichnisdienstes zu nutzen, bietet eRunbook für den Import der Benutzerdaten ein entsprechendes Interface über das LDAP Protokoll an. Die folgende Skizze verdeutlicht die Funktionsweise der Benutzerauthentifikation zwischen den verschiedenen Komponenten:



(Erläuterung siehe Folgeseite)

### Funktionsweise des LDAP Import Interface:

1. Der Browser schickt den Benutzernamen mit Passwort an den Webserver.
  2. Dieser kann den Benutzer entweder mit Hilfe des LDAP Interfaces gegen einen Verzeichnisdienst vergleichen oder mit der servereigenen Benutzerdatei htpasswd.
  3. Die eRunbook Software speichert den Benutzernamen ab.
  4. Rechteprüfungen innerhalb der eRunbook Software werden über die Datenbank geprüft.
- A. Das Perl Script `$NOVAHOME/product/tools/eRunbook_ldap_get_user.pl` fragt die Benutzernamen aus einem vorhandenen Verzeichnisdienst ab.
- B. Die gefundenen Benutzer werden in der entsprechenden htpasswd des Webserver mit dem Benutzernamen als Passwort gespeichert.
- C. Weiterhin werden die Benutzerdaten aus dem Verzeichnisdienst ausgelesen und in der eRunbook Datenbank gespeichert. Anschliessend werden die neu angelegten Benutzer entsprechend mit einer Usergruppe und einem benutzerspezifischen Menü verknüpft.



### 4.5.2 eRunbook\_ldap\_get\_user.pl

```
./eRunbook_ldap_get_user.pl <instanz> -b <base dn> -s <searchtype>
-h <host> -u <usergroup> [-m <menu>] [-y]
```

Name des Platzhalters	Beschreibung
instanz	eRunbook Instanzname, in welche die Benutzer importiert werden sollen
base dn	Base DN des LDAP Verzeichnisdienstes unter dem die Benutzerobjekte zu finden sind
searchtype	base   sub   one <i>base</i> : genau dieses Objekt <i>sub</i> : dieses Objekt und alles darunter <i>one</i> : eine Ebene unterhalb des BaseDNs
host	Hostname oder IP Adress des Servers auf dem der LDAP Verzeichnisdienst angeboten wird
usergroup	Name der Usergroup, mit der die importierten Benutzer verknüpft werden
menu	optional: Angabe eines benutzerspezifischen eRunbook Startmenüs

**ACHTUNG:** Bei der Ausführung des `eRunbook_ldap_get_user.pl` werden die über LDAP gelieferten und importierten Benutzerobjekte zusätzlich in einer CSV Datei geschrieben (`$NOVAHOME/instanz/<instanzname>/spool/user.csv` und `user_usergroup.csv`). Um bei einem erneuten Master Prozess nicht noch einmal alle LDAP Objekte separat importieren zu müssen, können aus diesen Dateien die Daten in die globale Master Datei `master_user.xls` übernommen werden.

**ACHTUNG:** Beim Import muß die Benutzergruppe zu welcher die importierten Benutzer gehören sollen angegeben werden. Bei der Angabe der Benutzergruppe muß der Präfix "usergroup\_" weggelassen werden, da die Software diesen Präfix automatisch setzt. (Bsp: Benutzergruppe "usergroup\_ro" wird zu "ro" als Benutzergruppe auf der Kommandozeile)

## 4.6 XML Import Interface

Mit Hilfe des XML Import Interfaces können Objekte und deren Attribute aus XML konformen Dateien in eRunbook importiert werden.

Programm: \$NOVAHOME/product/communication/xml.pl

Usage: ./xml.pl <instance> <xml-File> <Definition-File>

Parameter	Beschreibung
<instance>	Name der eRunbook Instanz
<xml-File>	Name der XML Datei, welche die zu importierenden Objekte enthält
<Definition-File>	Name der Definitions Datei, welche das Mapping der XML Datenstruktur auf die Datenstruktur von eRunbook beschreibt

Beispiel einer XML Datei, welche Rechnernamen enthält und deren zugehörige IP Adresse:

```
<test>
  <system name="rechner01">
    <chassis>big</chassis>
    <ip>10.0.0.1</ip>
  </system>
  <system name="rechner02">
    <chassis>medium</chassis>
    <ip>10.0.0.2</ip>
  </system>
</test>
```

Die Definitions Datei übernimmt die Zuordnung der Daten im XML File zu den entsprechenden Objektnamen mit dessen Attributnamen in der eRunbook Instanz. Die Datei wird im "ini-Format" erstellt. Beispiel einer Definitions Datei:

```
[osdomain]
object=/test/system
objectname=/system/@name
class_osdomain_field_chassis=/system/chassis
ip=/system/ip
view=static:view_osdomain
```

Der Sektionsname in eckigen Klammern (im obigen Beispiel "osdomain") entspricht dem Klassennamen der zu importierenden Objekte.

**HINWEIS:** Pro Sektion müssen mindestens drei Parameter angegeben werden: object, objectname und eine View (view, scanview oder defaultview)



Das Mapping im Beispiel ist folgendermaßen definiert:

	<b>Beschreibung</b>	<b>Beispiel</b>
<b>eRunbook Klasse</b> (object)	gibt den XPath zum Element an, innerhalb dessen alle Informationen für das zu importierende Objekt liegen. Der XPath muss absolut angegeben werden, d.h. inkl. root-Element. <b>Wichtig:</b> Alle weiteren XPath-Angaben sind relativ zu diesem Element	object=/test/system
<b>eRunbook Objekt</b> (objectname)	gibt den XPath zu dem Wert an, dem die Bezeichnung des Objektes entnommen werden kann. Der XPath ist relativ zu dem unter "eRunbook Klasse" definierten Element anzugeben.	als XML-Attribut (siehe Bsp): objectname=/system/@name oder als XML-Element: objectname=/system/name
<b>eRunbook Attribute</b>	gibt den XPath zu dem Wert an, dem der Wert des Attributes entnommen werden kann. Der XPath ist relativ zu dem unter "eRunbook Klasse" definierten Element anzugeben.	als XML-Element (siehe Bsp): class_osdomain_field_chassis =/system/chassis oder als XML-Attribut: class_osdomain_field_chassis =/system/@chassis
<b>eRunbook View</b>	gibt die zugehörige View des importierten Objektes an	view=static:view_osdomain

**HINWEIS:** Im Beispiel wird der View ein statischer Wert mit dem Schlüsselwort "static" zugewiesen. "static" bedeutet, dass alle importierten Objekte diese View zugewiesen bekommen ohne nochmal einzeln in der XML Datei aufgeführt zu werden. Die Tabelle auf der folgenden Seite listet alle Parameternamen auf, welche im Zusammenhang mit "static" benutzt werden können.

Mögliche Parameter, welche mit static kombiniert werden können sind:

<i>static Parametername</i>	<i>Beschreibung</i>
view	Name einer View. Als Trennzeichen bei der Angabe von mehreren Views dient das Komma ohne führendem oder nachfolgendem Leerzeichen.
scanview	Name einer Scanview. Als Trennzeichen bei der Angabe von mehreren Views dient das Komma ohne führendem oder nachfolgendem Leerzeichen.
defaultview	Name einer Defaultview
ip	IP Adresse des Rechners
time	Datum, Uhrzeit und (optional) Zeitzone für den Zeitstempel der Attributwerte der angegebenen Klasse Syntax: <Datum> <Uhrzeit> [Zeitzone] Beispiel: 01.01.2007 12:34:56 CET
acl	Eine ACL Liste für das Objekt der angegebenen Klasse. Falls dem Objekt keine ACL Wert zugewiesen wird, wird der Default ACL Wert benutzt. Als Trennzeichen bei der Angabe von mehreren ACL Werten dient das Komma ohne führendem oder nachfolgendem Leerzeichen.

Weiteres XPath Beispiel für eine Objekt Angabe:

`/test/system[@name = "rechner01"]/chassis`

Im Gegensatz zum XML-Beispiel oben, wo ein einfacher XPath `/test/system` genügt, werden hier nur bestimmte chassis-Elemente betrachtet: Das übergeordnete Element "system" muss ein Attribut mit dem Namen "name" haben und zusätzlich muss dem Systemnamen der Wert "rechner01" zugeordnet sein.

#### 4.7 Verzeichnisstruktur unter \$NOVAHOME

<i>Verzeichnisnamen</i>	<i>Beschreibung</i>
instanz	enthält die Verzeichnisse der einzelnen Instanzen (siehe auch 3.5.1)
log	beinhaltet die Logdateien aller eRunbook Komponenten
mysql	enthält die eRunbook Instanz Datenbanken der mySQL Datenbanksoftware
product	enthält die eRunbook Systemsoftware
software	beinhaltet Software von Drittanbietern, die vom eRunbook System zwingend oder optional benötigt werden
spool	in diesem Verzeichnis werden temporäre Daten gespeichert, die während der Laufzeit von eRunbook benötigt werden

#### 4.7.1 \$NOVAHOME/instanz/example

<i>Verzeichnisnamen</i>	<i>Beschreibung</i>
css	enthält die css Dateien für diese Instanz
etc	beinhaltet die Konfigurationsdateien der Instanz
export	
files	enthält ASCII Dateien der kundenspezifischen Daten. Wird vom Programm <code>eRunbook_admin dbread dbupdate</code> benutzt
html	enthält instanzspezifische HTML Dateien, auf die der Webserver zugreifen kann
master	enthält die Dateien für den Master Process
offline	enthält die Kommando- und Output Files, die bei einer Offline-Mode Kommunikation mit dem eRunbook Agent ausgetauscht werden
patternscripts	enthält eigene, instanzspezifische Rulescripts
picture	beinhaltet die instanzspezifischen Logos und Bilder
print	enthält instanzspezifische Konfigurationsdateien für die eRunbook Druckfunktion
scripte	kann z.B. Utility Programme für die Scan Funktion des eRunbook Systems enthalten
spool	wird für temporäre Dateien beim Ausdrucken genutzt
sql	enthält die Dateien, welche vom Programm <code>eRunbook_admin dbimport dbexport</code> erzeugt werden.
tmp	wird für unterschiedliche temporäre Dateien des eRunbook Systems genutzt

#### 4.7.2 \$NOVAHOME/product

<i>Verzeichnisnamen</i>	<i>Beschreibung</i>
admin	enthält Programme für die eRunbook Administration
agent	enthält die eRunbook Agent Software für den lokalen Server
applets	hier befinden sich die vom eRunbook System benötigten Java Applets
application	enthält die Komponenten des eRunbook Applikations Servers
cgi-bin	in diesem Verzeichnis befinden sich die cgi-Programme, auf welche der Webserver zugreift
communication	in diesem Verzeichnis sind die Programmdateien, welche für die Kommunikation des Servers mit dem Agent notwendig sind
db	enthält die Dateien, die für die Kommunikation mit der Datenbank benötigt werden
documentation	enthält die Dokumentationen zu eRunbook
etc	enthält die Konfigurationsdateien für den eRunbook Server
icons	enthält die Icons, die vom eRunbook Server benötigt werden
install	beinhaltet Installations- und Update Dateien
instance	wird für die Generierung von neuen Instanzen benötigt
javascript	hier befinden sich die vom eRunbook System benötigten Javascript Dateien
lib	enthält die Library Dateien, die vom eRunbook System benötigt werden
master	enthält Dateien, die für den Master Prozess benötigt werden
migrate	Migrationstools für das eRunbook System
patternscreens	enthält Rulescreens, welche im Zusammenhang mit "computed Attributes" benutzt werden können
perl	enthält einen symbolischen Link auf Perl, welches für den eRunbook Server benutzt wird
print	enthält die Layout Konfigurations Dateien, welche für die Druckfunktion von eRunbook benötigt werden
tools	enthält verschiedene Tools für das eRunbook System
update	enthält Updatedateien

## 4.8 Löschen einer eRunbook Instanz

Um eine eRunbook Instanz zu löschen sind nachfolgende Schritte auf der Kommandozeile des Betriebssystems durchzuführen.

**Achtung:** Alle Befehle sind mit dem Benutzernamen des eRunbook Servers durchzuführen (default Benutzereinstellung der Installationsroutine: erunbook).

Instanz stoppen:

```
$NOVAHOME/product/admin/eRunbook_admin stop <instanzname>
```

Instanzverzeichnis löschen:

```
cd $NOVAHOME/instanz/  
rm -rf <instanzname>
```

Anmelden und Löschen der Datenbank (z.B. MySQL):

```
# mysql -u<benutzername> -p  
password: <benutzerpasswort>  
mysql> drop database <instanzname>;  
mysql> quit;
```

Instanzeneintrag aus folgender Datei entfernen:

```
$NOVAHOME/product/etc/eRunbook.instance
```

Logfiles löschen:

```
$NOVAHOME/log/<instanzname>.log  
$NOVAHOME/log/wrapper.db.log.<Instanzname>
```

Instanzobjekt aus der eAdmin Instanz entfernen:

```
$NOVAHOME/product/admin/eRunbook_admin dbdelete eAdmin object <instanz  
name>
```

## 4.9 Existierende Klasse umbenennen

Um eine existierende Klasse umzubenennen sind die folgenden Schritte nötig:

Lösche alle Objekte und die Klasse mit dem alten Namen aus eRunbook:

```
#cd $NOVAHOME/product/admin;  
./eRunbook_admin dbdelete <instanzname> class <alter_Klassenname>
```

Nachdem die alte Klasse und deren Objekte aus der eRunbook Instanz entfernt sind, können die entsprechenden Worksheets mit Hilfe eines erneuten Masterlaufs neu angelegt werden.

## 4.10 Existierende Beschreibung (Description) einer Instanz ändern

Die Beschreibung (Description) zu einer Instanz kann durch die folgende Befehlszeile geändert werden:

```
#echo 'Description der Instanz' | db_admin_text_write.pl eAdmin -info  
<instanzname> class_instance_field_description/
```

## 4.11 Löschen des gesamten Inhaltes der Klasse “trash”

Objekte, die innerhalb der eRunbook Umgebung über die Activitybox gelöscht werden, werden nicht aus eRunbook entfernt, sondern der Klasse “trash” zugeordnet. Ein unwiederbringliches Löschen der Objekte aus dem “Papierkorb” kann mit folgendem Befehl durchgeführt werden.

```
#cd $NOVAHOME/product/admin;  
./eRunbook_admin dbdelete <instanzname> trash
```

## 4.12 Löschen der eRunbook Server Software

Mit den folgenden Schritten wird die installierte eRunbook Server Software vollständig gelöscht.

1) Stoppen des eRunbook Servers als Benutzer *root*:

```
#!/etc/init.d/eRunbook stop
```

2) Löschen des Verzeichnisses `$NOVAHOME`, in dem die eRunbook Software gespeichert ist.

3) Weiterhin existiert noch das Start/Stop Scripte ausserhalb des `$NOVAHOME` Verzeichnisses, welches ebenfalls gelöscht werden kann.

```
#rm /etc/init.d/eRunbook
```

Um eRunbook automatisch beim Hochfahren des Rechners zu starten, existieren weitere Dateien, die ebenfalls gelöscht werden können.

Zum Beispiel befinden sich bei SUSE die Dateien `KXXeRunbook` und `SXXeRunbook` in den entsprechenden Runlevel Verzeichnisse unterhalb von `/etc/init.d/`. Am wahrscheinlichsten sind die Dateien in den Verzeichnissen `/etc/init.d/rc3.d/` und/oder `/etc/init.d/rc5.d/` zu finden.

Nach dem Löschen dieser Dateien ist die eRunbook Software vollständig gelöscht.

### 4.13 Drucken von eRunbook Dokumenten

In der Sektion [server] der Datei \$NOVAHOME/product/etc/eRunbook.conf besteht die Möglichkeit die folgenden Parameter zu konfigurieren.

<b>Name des Parameters</b>	<b>Beschreibung</b>
use_html2ps	Entscheidet, ob eRunbook die alte Druckengine verwenden soll: yes: alte Druckengine no: neue Druckengine Default: yes
print_account_name	Name des Users mit dessen Berechtigung die Daten beim Drucken vom Webserver geladen werden. Default: eRunAdmin
print_account_pass	Password des Users mit dessen Berechtigung die Daten beim Drucken vom Webserver geladen werden. Default: go4it
print_header_<type>_left	steuert die Kopfzeile links
print_header_<type>_center	steuert die Kopfzeile in der Mitte
print_header_<type>_right	steuert die Kopfzeile rechts
print_footer_<type>_left	steuert die Fußzeile links
print_footer_<type>_center	steuert die Fußzeile in der Mitte
print_footer_<type>_right	steuert die Fußzeile rechts

Im Verzeichnis \$NOVAHOME/instanz/<instanzname>/print werden folgende Dateien ausgewertet:

<b>Name der Datei</b>	<b>Beschreibung</b>
eRunbook_<type>.book	Steuerdatei der Druckengine
eRunbook_<type>.title	HTML Datei, welche als Deckblattvorlage dient.

**HINWEIS:** Falls die Datei nicht vorhanden ist, wird als Standardvorlage die Datei im Verzeichnis \$NOVAHOME/product/print verwendet.

**eRunbook – Mastering Complexity**

Für den Platzhalter <type> in der Kopf-/Fußzeile, Deckblatt und Steuerdatei sind die folgenden Werte möglich:

<i>Wert</i>	<i>Beschreibung</i>
doku	Einstellungen für den DokuViewer
history	Einstellungen für den HistoryViewer
delta	Einstellungen für den CompareViewer
hostlist	Einstellungen für die Hostlisten
unknown	Einstellungen für sonstige Druckfunktionen

Mögliche Variablen für die Kopf-/Fußzeile sowie für das Deckblatt:

<i>Name</i>	<i>Beschreibung</i>	<i>Kopf-/Fußzeile</i>	<i>Deckblatt</i>
\$NOVAPROJECT	Wert aus: eRunbook.text, [global], project_name	X	X
\$NOVACUSTOMER	Wert aus: eRunbook.text, [global], customer_name	X	X
\$NOVAVERSION	Wert aus: eRunbook.text, [global], version	X	X
\$NOVACOPYRIGHT	Wert aus: eRunbook.text, [global], copyright	X	X
\$NOVADATE	Das Datum	X	X
\$NOVAHOST	Der Objektname	X	X
\$NOVAHOSTDESCRIPTION	Die Beschreibung des Objektes	X	X
\$NOVAVIEW	Die View, die gedruckt wird	X	X
\$NOVANOW	Der Zeitpunkt zu dem gedruckt wird	X	X
\$LOGOIMAGE	Das Firmenlogo aus: eRunbook.usermode, [nova], frmLogo	X	
\$PAGE	Die aktuelle Seite	X	
\$PAGES	Die Gesamtseitenzahl	X	



## eRunbook – Mastering Complexity

### *Mögliche Werte in der Datei eRunbook\_<type>.book:*

- embedfonts
- firstpage {p1,toc,c1}
- fontsize {4.0..24.0}
- fontspacing {1.0..3.0}
- headfontfont {courier{-bold,-oblique,-boldoblique},  
times{-roman,-bold,-italic,-bolditalic},  
helvetica{-bold,-oblique,-boldoblique}}
- headfootsize {6.0..24.0}
- headingfont {courier,times,helvetica}
- landscape
- left margin{in,cm,mm}
- links
- linkstyle {plain,underline}
- logoimage filename.{bmp,gif,jpg,png}
- no-compression
- no-duplex
- no-embedfonts
- no-encryption
- no-links
- no-localfiles
- no-numbered
- no-pscommands
- no-title
- no-toc
- pagelayout {single,one,twoleft,tworight}
- pagemode {document,outline,fullscreen}
- portrait
- right margin{in,cm,mm}
- size {letter,a4,WxH{in,cm,mm},etc}
- title
- tocfooter fff
- tocheader fff
- toclevels levels
- toctitle string
- top margin{in,cm,mm}
- webpage



## eRunbook – Mastering Complexity

*fff = heading format string; each 'f' can be one of:*

. = blank

/ = n/N arabic page numbers (1/3, 2/3, 3/3)

: = c/C arabic chapter page numbers (1/2, 2/2, 1/4, 2/4, ...)

1 = arabic numbers (1, 2, 3, ...)

a = lowercase letters

A = uppercase letters

c = current chapter heading

C = current chapter page number (arabic)

d = current date

D = current date and time

h = current heading

i = lowercase roman numerals

I = uppercase roman numerals

l = logo image

t = title text

T = current time

### ***Mögliche Fehlermeldungen:***

ERR001: No files were found or loadable.

ERR002: No pages were generated.

ERR003: The document contains too many files or chapters.

ERR004: HTMLDOC ran out of memory.

ERR005: The specified file could not be found.

ERR006: The comment contains a bad HTMLDOC formatting command.

ERR007: The image file is not in a known format.

ERR008: HTMLDOC was unable to remove a temporary file.

ERR009: HTMLDOC had an unspecified internal error.

ERR010: HTMLDOC encountered a networking error when retrieving a file via a URL.

ERR011: HTMLDOC was unable to read a file.

ERR012: HTMLDOC was unable to write a file.

ERR013: A HTML error was found in a source file.

ERR014: A table, image, or text fragment was too large to fit in the space provided.

ERR015: A hyperlink in the source files was unresolved.

ERR016: A header/footer string in the document contains a bad \$ command.

## 4.14 Scan Funktionalität

### 4.14.1 Online Agent Scan

Der Online Scan eines Rechners kann über das Webfrontend gestartet werden. Zusätzlich besteht die Möglichkeit den Online Scan über die Konsole zu starten. Das folgende Beispiel scannt einen Rechner mit dem Namen `solaris-host1` der Klasse `osdomain` im online Mode, der zu einer Instanz mit dem Namen `example` gehört.

```
#cd $NOVAHOME/product/admin  
#./eRunbook_admin onput example off -name solaris-host1 -class osdomain
```

### 4.14.2 Geplante nächtliche Online Scans aller Rechner

Der Befehl `eRunbook_admin` kann auch in die Cronjob Funktionalität des Betriebssystems integriert werden um beispielsweise eine automatisierte tägliche Ausführung des Scans durchzuführen. Das folgende Beispiel startet den online Scan aller Rechner der Klasse `osdomain`, die über die Suche `class_osdomain_search_all` gefunden werden, der eRunbook Instanz `example` nachts um 3:00 Uhr:

```
#  
# Crontab  
#  
0 3 * * * cd /opt/eRunbook/;. .bashrc; cd $NOVAHOME/product/admin;  
./eRunbook_admin onput example off -s class_osdomain_search_all
```

### 4.14.3 Offline Scan

Um die generierten Command Files und Value Files zu speichern, müssen die folgenden Verzeichnisse auf dem eRunbook Server vorhanden sein:

```
$NOVAHOME/instanz/eRunbook/offline/values/  
$NOVAHOME/instanz/eRunbook/offline/cmd/
```



## eRunbook – Mastering Complexity

Ein Offline Scan gliedert sich in die folgenden drei Schritte:

1. Generierung eines Command Files für das jeweilige Objekt
2. Ausführen des Command Files auf dem entsprechenden Rechner
3. Import des Value Files, welches die Ergebnisse enthält, in den eRunbook Server

### Schritt 1:

Um einen Rechner im Offline Modus zu scannen wird zuerst mit dem eRunbook\_admin Befehl das Command File erzeugt.

```
#cd $NOVAHOME/product/admin
#./eRunbook_admin offput <instance> off -n <hostname> -class <classname>
für einen einzelnes Objekt
oder
#./eRunbook_admin offput <instance> off -s <Suche>
für eine Menge von Objekten (Treffer einer Suche)
```

Das erzeugte Command File wird im Verzeichnis

`$NOVAHOME/instanz/eRunbook/offline/cmd/` gespeichert und erhält einen Dateinamen nach der folgenden Konvention:

`<Objektname>_<Kommunikations-Methode>_<IP-Adress of host>`

(Beispiel: ko-pc-55\_agent\_192.168.0.10)

### Schritt 2:

Das erzeugte Command File wird auf den zu scannenden Rechner transportiert und dort zur Ausführung gebracht. Hierbei ist darauf zu achten, daß das Command File von einem Benutzer gestartet wird, welcher die entsprechenden Rechte besitzt um eine fehlerfreie Ausführung zu gewährleisten. Die Ergebnisdateien werden in einem automatisch angelegtem Unterorder „result“ gespeichert.

**ACHTUNG:** Das Command File muß ggfls. nach Änderungen der Attributdefinitionen des Objektes neu erstellt und zum entsprechenden Rechner transportiert werden.

### Schritt 3:

Die erzeugte Ergebnisdatei muss zurück auf den eRunbook Server transportiert und in das Verzeichnis `$NOVAHOME/instanz/eRunbook/offline/values/` kopiert werden.

Mit dem folgenden Befehl werden die gewonnenen Daten in das System importiert:

```
#cd $NOVAHOME/product/admin
#./eRunbook_admin offget <instance> <value file>
```

Mit dem Import ist der Offline Scan beendet.



## 4.15 Apache Webserver

### 4.15.1 Start/Stop

Der Webserver kann über ein Script einzeln gestartet und gestoppt werden. Wegen der Dateiberechtigungen muß das Starten und Stoppen vom entsprechenden eRunbook Benutzer (default: erunbook) ausgeführt werden.

```
$NOVAHOME/product/admin/eRunbook_apache start  
$NOVAHOME/product/admin/eRunbook_apache stop
```

### 4.15.2 Benutzerauthentifizierung über Apache mit “htpasswd”

In der Datei httpd.passwd des Apacheservers sind die Zugangsdaten der Benutzer zum eRunbook Server hinterlegt.

```
$NOVAHOME/software/additional/etc/httpd.passwd
```

Beim Masterlauf werden die Benutzer mit aufgeführten Benutzernamen in der master\_user.xls Datei in die httpd.passwd Datei eingetragen. Als Kennwort wird jeweils der Benutzernamen verwendet. Das Kennwort kann vom Benutzer jederzeit nach dem Einloggen über die Weboberfläche mit einem Klick auf den dargestellten Benutzernamen im Menüframe geändert werden.

### 4.15.3 Änderung des SSL Zertifikates

Zum Ändern des SSL Zertifikates kann das folgende eRunbook Tool benutzt werden:

```
$NOVAHOME/product/tools/eRunbook_gen_ssl_certificate.sh
```

Während der Ausführung des Skriptes erscheinen einige Hinweise und im Anschluss daran die Abfragen für das SSL Zertifikat für den Webbrowser des eRunbook Servers:

```
product/tools> ./eRunbook_gen_ssl_certificate.sh

writing RSA key
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:DE
State or Province Name (full name) [Some-State]:Germany
Locality Name (eg, city) []:Hundsdorf
Organization Name (eg, company) [Internet Widgits Pty Ltd]:nova ratio AG
Organizational Unit Name (eg, section) []:eRunbook
Common Name (eg, YOUR name) []:ko-pc-112
Email Address []:

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:

Signature ok
subject=/C=DE/ST=Germany/L=Hundsdorf/O=nova ratio AG/OU=eRunbook/CN=ko-pc-112
Getting Private key

Please restart Webserver now
```

Nach der Generierung des Zertifikates muß der Webserver neu gestartet werden.

**HINWEIS:** Beim Ausfüllen der Felder ist ein Punkt besonders zu beachten. Der "Common Name" muß den kompletten Namen, den Fully Qualified Domain Name (FQDN) des Servers enthalten! Der Internetbrowser vergleicht den entsprechenden Eintrag im Zertifikat mit dem Hostnamen in der URL und gibt bei Nichtübereinstimmung eine Warnung aus. Durch die Eingabe des richtigen Hostnamens kann dem späteren eRunbook Benutzer diese Warnung gespart werden.

## 4.16 mySQL Datenbanksystem

### 4.16.1 Start/Stop

Die mySQL Datenbank kann über ein Script einzeln gestartet und gestoppt werden. Wegen der Dateiberechtigungen muß das Starten und Stoppen vom entsprechenden eRunbook Benutzer (default: erunbook) ausgeführt werden.

```
$NOVAHOME/product/admin/eRunbook_mysql start  
$NOVAHOME/product/admin/eRunbook_mysql stop
```

### 4.16.2 Backup

**HINWEIS:** Für ein Backup der Datenbank sind die entsprechenden Tools und Dokumentationen des Datenbankherstellers zu benutzen.

### 4.16.3 Standard Tabellengrößenbegrenzung ändern

Die mit eRunbook bis einschließlich Version 3.0 ausgelieferte mySQL Konfiguration begrenzt die Größe der einzelnen Tabellen auf 4 GB. Wird diese Grenze überschritten, meldet mySQL einen Fehler. Ab der eRunbook Version 3.1 wurde dieser Wert erhöht.

Fehlermeldung: siehe <http://dev.mysql.com/doc/refman/4.1/en/full-table.html>

Lösungsvorschlag:

In der Datei \$NOVAHOME/software/mandatory/etc/mysql.cnf den Wert `myisam_data_pointer_size` auf z.B. 5 erhöhen.

## 4.17 Oracle Datenbanksystem

### 4.17.1 Kompatibilität Oracle Client zu Oracle Server Versionen

Seit eRunbook Version 3.x liegt der Software ein eigener Oracleclient bei. Es handelt sich hierbei um den von Oracle herausgegebenen Instantclient (Version 10). Die Abwärtskompatibilität des Clients wurde mit einem Oracleserver Version 9.2 getestet.

### 4.17.2 Voraussetzungen für den Betrieb

- Oracle Version: 9.x (siehe auch vorherigen Abschnitt)
- Pro eRunbook Instanz ist entweder ein Schema oder eine Oracle Instanz anzulegen
- \$NOVAHOME/.TNS\_ADMIN/tnsname.ora
- \$NOVAHOME/.TNS\_ADMIN/sqlnet.ora
- Folgende Informationen müssen bekannt sein: Oracle Benutzer (incl. Passwort), Namen der Tablespaces und der TNS Name

### 4.17.3 Zugriffsrechte auf die Oracle Datenbank

Die eRunbook Software muss während der Installation und während des Betriebes folgende Benutzerrechte für den Oraclezugriff haben:

<b>Objekt</b>	<b>anlegen/hinzufügen</b>	<b>ändern</b>	<b>löschen</b>
Tabelle	X	X	X
Index	X	X	X
Datensatz	X	X	X

#### 4.17.4 Oracle Tablespaces

Alle eRunbook Daten können in einem Tablespace gespeichert werden. Allerdings kann eRunbook seine Daten auf bis zu 3 Tablespaces verteilen. Für eine Installation empfehlen sich die folgenden Tablespaces:

<i>Type</i>	<i>Daten</i>
customize	Objekte, Views, Attribute, Regeln, usw.
history	Attributwerte zu einem Objekt
index	Indexe

**HINWEIS:** Die Größe der Tablespace hängt von dem erwarteten Volumen ab. Bitte beachten Sie hier das Dokument 'eRunbook Calculator'.

#### 4.17.5 eRunbook.conf Einstellungen für die Oracle Anbindung

Die folgenden Parameter können/müssen in der eRunbook.conf Datei angepasst werden.

<i>Option</i>	<i>Wert</i>	<i>Optional</i>
tablespace	1	X
tablespace_cmd	TABLESPACE	X
database_tablespace_customize	<NAME>	X
database_tablespace_history	<NAME>	X
database_tablespace_index	<NAME>	X
db_user	<USER>	
db_passwd	<PASSWORD>	
db_name	<TNS NAME aus tnsnames.ora>	
db_generic	0	
db_mod	Oracle	
db_connectionAdd	<leer>	
db_driver	oci	

## 4.17.6 Anleitung zur Integration der Oracle Datenbank

1. Anlegen einer neuen eRunbook Instanz über die Administration
2. Stoppen des eRunbook Servers:

```
cd $NOVAHOME/product/admin  
./eRunbook_admin stop
```

3. Bearbeiten der Datei \$NOVAHOME/instanz/<INSTANZNAME>/etc/eRunbook.conf in der Sektion [db] mit den folgenden obligatorischen Parametern:

```
db_user=<Name des Users in der Datenbank>  
db_passwd=<Passwort des Users in der Datenbank>  
db_name=<TNS Name aus tnsnames.ora>  
db_generic=0  
db_mod=Oracle  
db_connectionAdd=  
db_driver=oci
```

4. Setzen der Umgebungsvariable NLS\_LANG:

Bearbeiten Sie die Datei \$NOVAHOME/.bashrc und setzen Sie die Einstellungen auf den gleichen Wert, der in der Datenbank verwendet werden wird. NLS\_LANG muss in beiden Fällen identisch sein! (z.B. NLS\_LANG=german)

5. Initialisieren der eRunbook Software:

```
cd $NOVAHOME/product/admin  
./eRunbook_admin start  
./eRunbook_admin dbinit <INSTANZNAME> -y  
./eRunbook_admin masterinit <INSTANZNAME> -y
```

6. Restart der eRunbook Software:

```
cd $NOVAHOME/product/admin  
./eRunbook_admin stop  
./eRunbook_admin start
```

7. Ende der Oracle Einbindung

## 4.18 Datenbank allgemein

### 4.18.1 db\_thin\_out.pl

In eRunbook wird jeder neue Wert für ein Attribut als neuer Eintrag in der Datenbanktabelle history\_t mit einem Timestamp gespeichert. Mit eRunbook wird ein Programm ausgeliefert, mit dessen Hilfe historische Datensätze, die nicht mehr benötigt werden, unwiderruflich aus der Datenbank gelöscht werden können.

**ACHTUNG:** Bitte führen Sie vor dem Benutzen dieses Scriptes unbedingt eine Datensicherung durch.

Programm: \$NOVAHOME/product/db/db\_thin\_out.pl

Usage: ./db\_thin\_out.pl <instance> <from> <to> <first|last|none>  
 [-s <search>|-o <object>] [-a <attribute>] [-y]

Parameter	Beschreibung	Beispiel
<instance>	Name der eRunbook Instanz	eInventory
<from>	Datum (Format: DD.MM.YYYY)	01.01.2006 oder mit Uhrzeit '01.01.2006 12:34:56' (Datum mit Uhrzeit in <i>einfachen</i> Anführungszeichen)
<to>	Datum (Format: DD.MM.YYYY)	31.01.2006 oder mit Uhrzeit '01.01.2006 12:34:56' (Datum mit Uhrzeit in <i>einfachen</i> Anführungszeichen)
<first last none>	Erhalte den ersten (first), letzten (last) oder keinen (none) historischen Datensatz des gewählten Zeitraums	last
<i>optionale Parameter</i>		
-s <search>	Angabe einer Search, welche die Menge der Objekte einschränkt	class_osdomain_search_offline
-o <object>	Angabe eines Objektes, auf das das Löschen beschränkt wird	mailserver01
-a <attribute>	Angabe eines Attributes, auf das das Löschen beschränkt wird	class_osdomain_field_description
-y	unterdrückt die Sicherheitsnachfrage für das Löschen eines Datensatzes	

**HINWEIS:** Das Programm *db\_thin\_out.pl* berücksichtigt beim Löschen auch die automatisch generierten Job Objekte. Sollten alle Daten auf die sich ein Job bezieht durch einen *thin\_out* Lauf gelöscht werden, so wird der entsprechende Job ebenfalls gelöscht.

### **Beispiele:**

Lösche alle Einträge vom Januar 2006 mit Ausnahme des letzten Eintrages:

```
#cd $NOVAHOME/product/db;  
./db_thin_out.pl eInventory 01.01.2006 31.01.2006 last
```

Lösche alle Attributwerte des Attributes “class\_osdomain\_field\_description” vom Januar 2006:

```
#cd $NOVAHOME/product/db;  
./db_thin_out.pl eInventory 01.01.2006 31.01.2006 none -a  
class_osdomain_field_description
```

Lösche alle Attributwerte des Objektes “mailserver01” mit Ausnahme der ersten vom Januar 2006:

```
#cd $NOVAHOME/product/db;  
./db_thin_out.pl eInventory 01.01.2006 31.01.2006 first -o mailserver01
```

Das Programm *db\_thin\_out.pl* kann vom Administrator vielfältig eingesetzt werden.

- Der Aufruf des Programms kann in eRunbook über einen Menüeintrag eingebunden werden.
- Das Programm kann zur automatisierten Datenreduzierung über einen Cron Prozess des Betriebssystems integriert werden.

## 5 Anhang

Für einen optimalen Betrieb der eRunbook Instanz sollten folgende Rahmenbedingungen beachtet werden.

### 5.1 Hardware Sizing

Das Hardware Sizing eines eRunbook Servers hat als Ziel die Definition einer wirtschaftlichen Hardwareplattform für den Einsatz als eRunbook Server. Das Hardwaresizing wird von drei wesentlichen Faktoren bestimmt:

Größe der der eRunbook Datenbank

Schreibperformance der eRunbook Datenbank

Lesepformance der eRunbook Datenbank

Als Hilfsmittel für das Hardware Sizing steht der eRunbook Calculator als XLS Datei zur Verfügung, welcher das Sizing mit Hilfe von drei Gruppen von Objekten systematisiert:

Gruppe 1: Server und Stagesystem

Gruppe 2: Netzwerkkomponenten (Router und Switch)

Gruppe 3: PC-Systeme und Netzwerkdrucker

### 5.2 Datenbankgröße

Die Größe der Datenbank ergibt sich aus der Multiplikation der Anzahl der Objekte mit der pro Objekt erfassten Datenmenge mal Zeitspanne, für welche diese Daten aufbewahrt werden sollen.

Diese Nettogröße der Datenbank muss mit einem Faktor multipliziert werden, um auf die Bruttogröße der Datenbank zu kommen. Dieser Bruttofaktor berücksichtigt z.B. Datenbankindizes, Blockgrößen, etc. und ist abhängig vom eingesetzten Datenbankmanagementsystem.

Im eRunbook Calculator werden für die Objektgruppen jeweils die Anzahl der Objekte, die Anzahl der Attribute und die durchschnittliche Größe eines Attributwertes angegeben. Für die Kalkulation hat sich eine durchschnittliche Attributgröße von ca. 2kB bewährt.

Die Veränderungsquote der Attributwerte muss in jeder eRunbook Umgebung gemessen werden. Nur bei einer Änderung eines Attributwertes gegenüber der letzten Erfassung speichert der eRunbook Server diese Information in der Datenbank.

Als erste Schätzung kann eine Veränderungsquote von 20% angenommen werden.

Die Verweildauer der Informationen in der Datenbank hängt neben den Anforderungen an das eRunbook System wesentlich von der Ausdünnungsstrategie ab. Für die Kalkulation wird von einer vollständigen Informationshistorie von 24 Monaten ausgegangen.

In der Praxis kann durch eine geschickte Ausdünnung der Informationshistorie mit der gleichen Datenmenge auch wesentlich ältere Informationen in der Datenbank vorgehalten werden.

### 5.3 Schreibperformance

Die Schreibperformance ist ein Maß für die Leistungsfähigkeit der Hardwareplattform für die Verarbeitung der erfassten Daten im Scan Interface sowie im CSV Interface.

Pro Hardwaremodell wird dazu ein Durchsatzwert in MB/s ermittelt, welcher durch Messungen beim Einsatz der jeweiligen Hardwareplattform als eRunbook Server entsteht. Die Datenmenge pro Objekt dividiert durch diesen Durchsatzwert ergibt die Verarbeitungsdauer pro Objekt (**D**):

$$D = \frac{S}{A}$$

D: Verarbeitungsdauer pro Objekt  
 S: durchschnittlich geschriebene, d.h. geänderte Datenmenge pro Objekt  
 A: Schreibdurchsatz der Hardwareplattform in MB/s

Neben diesem Schreibdurchsatz bestimmt die Anzahl paralleler Schreibprozesse wesentlich die Verarbeitungsdauer. Dabei gilt:

$$t_2 = \frac{O}{W} * D$$

t<sub>2</sub>: minimale Verarbeitungsdauer für die erfassten Daten  
 O: Anzahl erfasster Objekte  
 W: Anzahl paralleler Schreibprozesse  
 D: Verarbeitungsdauer pro Objekt

Aus dieser Gesetzmäßigkeit ergeben sich im wesentlichen zwei Strategien für die Minimierung der Gesamtverarbeitungsdauer:

**Verkürzung der Verarbeitungsdauer pro Objekt**

Die Verarbeitungsdauer pro Objekt nimmt mit der Anzahl der erfassten Attribute und mit der Anzahl der den Attributen zugeordneten Rules zu und kann nur bedingt durch die Parametrisierung des eRunbook Systems verkürzt werden.

Durch den Einsatz leistungsfähiger Prozessoren, d.h. in diesem Fall höher getakteter Prozessoren, wird die Verarbeitungsdauer pro Objekt ebenfalls verkürzt.

**Erhöhung der Anzahl paralleler Schreibprozesse**

Durch die Erhöhung der Anzahl paralleler Schreibprozesse kann die Gesamtverarbeitungsdauer deutlich verkürzt werden, sofern das Datenbankmanagementsystem und das Betriebssystem des eRunbook Servers keinen Engpaß darstellen.

In typischen eRunbook Umgebungen kommen zur Gesamtverarbeitungsdauer noch die Vorverarbeitung für Datenschnittstellen und ggf. die Berechnung von den Einzelobjekten unabhängiger Informationen hinzu, so daß die Gesamtverarbeitungsdauer sich wie folgt definiert:

$$t_g = t_1 + t_2 + t_3$$

t<sub>g</sub>: nächtliche Verarbeitungsdauer inklusive Vor- und Nachverarbeitung  
 t<sub>1</sub>: Verarbeitungsdauer für die Datenschnittstellen etc.  
 t<sub>2</sub>: minimale Verarbeitungsdauer für die erfassten Daten  
 t<sub>3</sub>: Verarbeitungsdauer für die Nachverarbeitung



Zur Verdeutlichung dient das folgende Beispiel mit 1000 Objekten, für welche Daten erfasst werden:

<i>Parameter</i>	<i>Wert</i>
Anzahl Objekte ( <b>O</b> )	1000 Stück
Verarbeitungsdauer pro Objekt ( <b>D</b> )	30 Sekunden
Anzahl paralleler Schreibprozesse ( <b>W</b> )	5 Stück

Die minimale Verarbeitungsdauer beträgt also 6000 Sekunden bzw. 1,67 Stunden:

$$t_2 = \frac{1000}{5} * 30 s = 6000 s$$

Bei einer Erhöhung der Anzahl der parallelen Schreibprozesse auf 8 Stück ergibt sich eine minimale Verarbeitungsdauer von 3750 Sekunden bzw. 1,04 Stunden:

$$t_2 = \frac{1000}{8} * 30 s = 3750 s$$

Um die gleiche Verbesserung durch eine Verkürzung der Verarbeitungsdauer pro Objekt zu erreichen, müsste diese Zeit auf 18,75 Sekunden verkürzt werden. Diese Verkürzung der Verarbeitungszeit pro Objekt um ca. 38% ist in der Praxis jedoch nicht realistisch:

$$t_2 = \frac{1000}{5} * 18,75 s = 3750 s$$

**Hinweis:** Je mehr Zeit dem eRunbook Server für die Gesamtverarbeitungsdauer  $t_g$  eingeräumt wird, um so geringer fallen die Anforderungen an die Leistungsfähigkeit der eRunbook Serverhardware aus. Für die nächtliche Datenerfassung ist daher u.U. die Startzeit der Datenerfassung, d.h. die Anlieferung der Daten, bedeutsamer als die Parallelisierung der Verarbeitung.

## 5.4 Leseperformance

Die Leseperformance drückt die Leistungsfähigkeit der Hardwareplattform für die Darstellung von Informationen in der Runtime Umgebung aus.

Neben der Leistungsfähigkeit der Hardwareplattform ist hierbei die Parametrisierung der Runtime Umgebung von Bedeutung. Dies wird mit den beiden folgenden Beispielen deutlich.

### 5.4.1 Leseperformance der Suche

Die Durchführung einer Suche in der Runtime Umgebung besteht aus zwei Teilen:

Bestimmung der Objektmenge

Darstellung der Objektmenge in Form einer Liste

Für die Bestimmung der Objektmenge werden alle konfigurierten Suchzeilen ausgeführt und ihre Ergebnisse danach mit Hilfe des logischen Ausdrucks der Suche zu einem Endergebnis verarbeitet. Die meisten Suchoperation nutzen hierzu einen Applikationscache im Hauptspeicher. Für die Parametrisierung der Suche ergeben sich daraus die folgenden Aspekte:

Nutzung von verschachtelten Suchen zur Einschränkung der Datenmenge

Vermeidung unnützer Suchzeilen

Bereitstellung von ausreichend Hauptspeicher auf dem eRunbook Server

Die Darstellung als Liste erfolgt gemäß der Listendefinition und wird für jede Darstellung neu berechnet. Für die Parametrisierung der Listen gelten die folgenden Punkte:

Minimierung der Spaltenanzahl

Minimierung von Spalten mit Attributtypen, deren Informationen on-the-fly berechnet werden

Nutzung des Blockcount Parameters für die Limitierung der Anzahl gleichzeitig dargestellter Zeilen in der Liste

**Achtung:** Die Parametrisierung von eRunbook ermöglicht die Definition von Suchen und Listen, deren Darstellung lange dauern kann, z.B. die Suche nach allen Objekten einer Klasse, Blockcount = 0 und Darstellung in einer Liste mit 20 Spalten, von denen 10 Attribute vom Typ treepath enthalten.

## 5.4.2 Leseperformance des DocViewers

Die Darstellung einer View im DocViewer der Runtime Umgebung besteht aus dem Einlesen der Dokumentenstruktur und dem Lesen aller der Struktur zugeordneten Attributwerte für das dargestellte Objekt. Für die Darstellung dieser Attributwerte muss je nach Attributtyp unterschieden werden:

<b>Attribute Type</b>	<b>Performance-einfluß</b>	<b>Beschreibung</b>
html	gering	Übergabe des Attributwertes an den Webbrowser
system	gering	Übergabe des Attributwertes an den Webbrowser mit Ergänzung durch PRE Tags
date	gering	Aufbereitung des Datumswertes als Datumsstring und Übergabe an den Webbrowser
compare	gering	Aufbereitung der vorberechneten Veränderungsinformation als Ampel und Übergabe an den Webbrowser
link	gering	Übergabe des Links an den Webbrowser
binary	yes	Attributwert beinhaltet Binärzeichen, die Darstellung erfolgt durch ein Icon mit Link, welcher die Binärdatei zur Darstellung an den Webbrowser überträgt
treepath	mittel	Objektsuche über die Beziehungen entlang eines Klassenpfades
search	hoch	Ausführung der Suche und anschließende Darstellung als Liste

Damit ergeben sich für die Leseperformance des DocViewers die folgenden Aspekte:

Minimierung der einzelnen Views, Nutzung mehrerer Views

Vermeidung der Komplettdarstellung im Viewer

Vermeidung von Attributen vom Typ `search` innerhalb von Views

Zur Verdeutlichung dient das folgende Beispiel eines Handbuchs für einen Service, dem 100 Rechner zugeordnet sind.

Es wird eine View parametrisiert, in welcher alle Informationen enthalten sind. D.h. in der View sind sowohl die manuellen Textteile des Handbuchs enthalten (50 Attribute vom Typ `html`) als auch die Rechnerinformationen. Für die Rechnerinformationen werden 100 Attribute vom Typ `search` parametrisiert, welche die Rechner zum Service suchen und die 100 unterschiedlichen Attribute der Rechner darstellen.



## eRunbook – Mastering Complexity

Für die Darstellung dieser View ergibt sich damit die folgende Anzahl von Attributwerten, welche aus der Datenbank gelesen werden müssen:

$$A_{view} = A_{simple} + A_{dynamic}$$

- $A_{view}$ : Anzahl Attributwerte für die View
- $A_{simple}$ : Anzahl einfacher Attributwerte in der View
- $A_{dynamic}$ : Anzahl dynamischer Attributwerte in der View

Für die dynamischen Attribute, d.h. für die Suchen gilt:

$$A_{dynamic} = A_{search} * A_{objectpersearch} * A_{colperlist}$$

- $A_{dynamic}$ : Anzahl dynamischer Attributwerte in der View
- $A_{search}$ : Anzahl Suchen in der View
- $A_{objectpersearch}$ : Anzahl Objekte als Treffermenge der Suchen
- $A_{colperlist}$ : Anzahl Spalten mit Attributwerten in den Listen für die Darstellung der Suchergebnisse

In dem Beispiel ergibt sich also, daß für eine Komplettdarstellung dieser View 10100 Attributwerte aus der Datenbank gelesen und im Webbrowser dargestellt werden müssen:

$$A_{dynamic} = 100 * 100 * 1 = 10000$$

$$A_{view} = 100 + 10000 = 10100$$

**Hinweis:** Die Parametrisierung der Views sollte die oben genannten Aspekte der Leseperformance berücksichtigen. Es ist durch die Parametrisierung einer View möglich, auf jeder Hardwareplattform eine nicht zufriedenstellende Leseperformance aus Sicht des Anwenders zu erhalten. Die goldene Regel für die Views, welche interaktiv genutzt werden, lautet: Parametrisiere nur die Menge an Informationen in eine View, welche der Anwender nach einem Klick auch lesen wird, und lege für den zweiten Klick des Anwenders eine weitere View an.