

Beratung und Support
Technische Plattform
Support-Netz-Portal



paedML® – stabil und zuverlässig vernetzen

Anleitung

**Dateien, Ordner ohne Owner.
BenAdmin gelöscht? Was nun?**

Stand 04.10.2017

paedML® Novell

Version: 4.x

Impressum

Herausgeber

Landesmedienzentrum Baden-Württemberg (LMZ)
Support-Netz
Rotenbergstraße 111
70190 Stuttgart

Autoren

der Zentralen Expertengruppe Netze (ZEN),
Support-Netz, LMZ

Hubert Bechtold
Holger Dzeik
Stefan Falk
Ulrich Frei
Carl Heinz Gutjahr
Friedrich Heckmann
Uwe Labs
Alfred Wackler

Endredaktion

Uwe Labs

Bildnachweis Titelbilder:

Thinkstock

Weitere Informationen

www.support-netz.de
www.lmz-bw.de

Änderungen und Irrtümer vorbehalten.

Veröffentlicht: 2017

© Landesmedienzentrum Baden-Württemberg

Inhaltsverzeichnis

1.	BenAdmin gelöscht	5
1.1	Testumgebung	5
1.2	Löschen des BenAdmin	5
1.3	Reparatur	6
2.	Allgemeine Benutzer gelöscht.....	7
3.	Wann tätig werden ?.....	8
4.	Schluss	9
Anhang A(Filr).....		9

Vorwort

Werden Benutzer, aber nicht deren Dateien und/oder Ordner gelöscht, haben eben diese Dateien/Ordner keinen gültigen Besitzer mehr.

Dies kann zu Problemen führen. So zum Beispiel im *Filr*. Der versucht nämlich nun den jeweiligen Creator (Erzeuger, Besitzer (Owner)) zu finden, was ihm aber nicht gelingt. Dies schreibt der *Filr* in seine Log-Datei */var/opt/novell/filr/log/famtd.log*. Bei vielen Usern geschieht dies massenhaft, was den *Filr* unter hohe Last setzt. (Siehe auch Anhang A)

Möglicherweise gibt es noch weitere unerwünschte „Nebenwirkungen“.

Werden „normale“ Benutzer (Lehrer/Schüler) gelöscht, so werden i.d.R. deren Home-Verzeichnisse auch gelöscht oder nach „Abgang“ verschoben. Dies macht keine Probleme, da der *Filr* nicht nach gelöschten Dateien bzw. in Abgang sucht.

Haben die Benutzer aber in Projektverzeichnissen Dateien/Ordner besessen, die nicht gelöscht wurden, so haben diese nun keinen Owner mehr.

Ein weiteres Problem tritt auf, wenn Sie den speziellen Benutzer BenAdmin löschen.

Normalerweise haben Sie Ihre User per *Schulkonsole/Blimport* oder in Einzelfällen per *iManager* erzeugt. Dabei sind Sie auf einer Arbeitsstation im Netz eingeloggt, meist wohl als BenAdmin Ihrer Schule.

In jedem Fall wird beim Erzeugen eines Users beim angelegten zugehörigen Homeverzeichnis ein Creator (=Ersteller), also der Besitzer (Owner) dieses Verzeichnisses eingetragen. Dies ist der während des User-Erzeugens eingeloggte Benutzer, also i.d.R. der BenAdmin (oder vielleicht auch der admin).

Eigentlich ist es deswegen nicht zweckmäßig den BenAdmin aus dem System löschen. Auch dann nicht, wenn der dieses Amt ausführende Kollege die Schule verlässt. (Wenn nötig können Sie ja die persönlichen Daten aus dem Homeverzeichnis des BenAdmin löschen.)

Löschen Sie aber trotzdem den BenAdmin, so haben anschließend alle Homeverzeichnisse der vom BenAdmin erzeugten User einen undefinierten Creator/Owner-Eintrag.

Für Owner-Probleme, die möglicherweise aus solchen Lösch-Aktionen entstehen, gibt es glücklicherweise verschiedene Lösungen. Wir beschreiben dies mit Hilfe des Programms *nsschown*, das auf dem GServer03 standardmäßig vorhanden ist. Dies lässt sich als Kommandozeile an der Server-Konsole oder leichter per SSH mit PuTTY ausführen.

Man kann mit Hilfe von *nsschown* sowohl

- gezielt verloren gegangene Owner-Einträge reparieren, die durch Löschen eines bestimmten Benutzers, z.B. des BenAdmin, entstanden sind, (→ Kap. 1)
- als auch generell verloren gegangene Owner-Einträge reparieren, die durch Löschen vieler Benutzer entstanden sind. (→ Kap. 2)

1. BenAdmin gelöscht

Das hier beschriebene Verfahren lässt sich natürlich auch auf andere einzelne gelöschte Benutzer anwenden. Wem das zu aufwändig ist, kann auch die mehr „globale“ Methode verwenden, wie sie in Kap. 2 beschrieben ist.

1.1 Testumgebung

Für den Fall des BenAdmin haben wir beim Testen eine kleine Testumgebung auf einem Test-GServer03 zusammengestellt. Diese sieht so aus:

Im Context *lehrer.benutzer.S01.SCHULEN.ml3* bzw. im Filesystem *DOCS\S01\home\lehrer* haben wir zwei User: *SpechtB-S01* und *SperlingH-S01*.

Im Homeverzeichnis von *SpechtB-S01* gibt es eine Datei *Test.txt* mit dem Owner *SpechtB-S01* und im Homeverzeichnis von *SperlingH-S01* gibt es ein Unterverzeichnis *public_html* mit dem Owner *BenAdmin-S01.BenAdmins.Verwalter.Benutzer.S01.SCHULE.ml3*.

Loggt man sich am GServer03 als *root* (z.B. per PuTTY oder an der Konsole) ein, lässt sich dies mit dem Programm *nsschown* auflisten:

```
gserver03:~ # nsschown -l DOCS:S01/home/lehrer -a
```

Die Ausgabe sieht dann so aus (an Ihrer Schule natürlich viel viel länger):

```
Path::DOCS:S01/home/lehrer| Owner
DN::.CN=admin.OU=Server.OU=DIENSTE.O=ml3.T=SCHULBAUM03.| Owner GUID::c47579bc-c8c2-41f3-
98-be-bc7975c4c2c8
Path::DOCS:S01/home/lehrer/SpechtB-S01| Owner DN::.CN=BenAdmin-
S01.OU=BenAdmins.OU=Verwalter.OU=Benutzer.OU=S01.OU=SCHULEN.O=ml3.T=SCHULBAUM03.| Owner
GUID::e1b3b28c-6936-49e8-d1-ad-8cb2b3e13669
Path::DOCS:S01/home/lehrer/SperlingH-S01| Owner DN::.CN=BenAdmin-
S01.OU=BenAdmins.OU=Verwalter.OU=Benutzer.OU=S01.OU=SCHULEN.O=ml3.T=SCHULBAUM03.| Owner
GUID::e1b3b28c-6936-49e8-d1-ad-8cb2b3e13669
Path::DOCS:S01/home/lehrer/SperlingH-S01/public_html| Owner DN::.CN=SperlingH-
S01.OU=Lehrer.OU=Benutzer.OU=S01.OU=SCHULEN.O=ml3.T=SCHULBAUM03.| Owner GUID::a6165b6e-
8f32-4550-bd-b3-6e5b16a6328f
Path::DOCS:S01/home/lehrer/SpechtB-S01/Test.txt| Owner DN::.CN=SpechtB-
S01.OU=Lehrer.OU=Benutzer.OU=S01.OU=SCHULEN.O=ml3.T=SCHULBAUM03.| Owner GUID::ea91faae-
9257-469d-0e-a9-aefa91ea5792

Operation Succeeded.
```

Beachten Sie dabei die GUID des BenAdmin-S01: *e1b3b28c-6936-49e8-d1-ad-8cb2b3e13669*

1.2 Löschen des BenAdmin

Nun wurde unglücklicherweise der BenAdmin-S01 aus dem eDirectory gelöscht.

Danach sieht das Ergebnis mit dem List-Befehl

```
gserver03:~ # nsschown -l DOCS:S01/home/lehrer -a
```

so aus:

```
Path::DOCS:S01/home/lehrer| Owner
DN::CN=admin.OU=Server.OU=DIENSTE.O=ml3.T=SCHULBAUM03.| Owner GUID::c47579bc-c8c2-41f3-
98-be-bc7975c4c2c8
Path::DOCS:S01/home/lehrer/SpechtB-S01| Owner DN::| Owner GUID::e1b3b28c-6936-49e8-d1-ad-
8cb2b3e13669
Path::DOCS:S01/home/lehrer/SperlingH-S01| Owner DN::| Owner GUID::e1b3b28c-6936-49e8-d1-ad-
8cb2b3e13669
Path::DOCS:S01/home/lehrer/SperlingH-S01/public.html| Owner DN::CN=SperlingH-
S01.OU=Lehrer.OU=Benutzer.OU=S01.OU=SCHULEN.O=ml3.T=SCHULBAUM03.| Owner GUID::a6165b6e-
8f32-4550-bd-b3-6e5b16a6328f
Path::DOCS:S01/home/lehrer/SpechtB-S01/Test.txt| Owner DN::CN=SpechtB-
S01.OU=Lehrer.OU=Benutzer.OU=S01.OU=SCHULEN.O=ml3.T=SCHULBAUM03.| Owner GUID::ea91faae-
9257-469d-0e-a9-aefa91ea5792
Operation Succeeded.
```

Beachten Sie nun: Wo oben der BenAdmin-S01 zu sehen war, fehlt dieser jetzt.

Aber: Seine GUID ist noch da, nämlich: e1b3b28c-6936-49e8-d1-ad-8cb2b3e13669.

1.3 Reparatur

Dies ermöglicht uns nun, in allen Verzeichnissen und Dateien, die als Creator/Owner diese GUID haben, diese durch einen existierenden User zu ersetzen. Wir schlagen hier den *admin* vor. (Sie können auch einen in der Zwischenzeit neu angelegten BenAdmin nehmen.)

Dies geht mit dem Befehl:

```
gserver03:~ # nsschown -l DOCS:S01/home/lehrer -g e1b3b28c-6936-49e8-d1-ad-
8cb2b3e13669 -n .admin.server.dienste.ml3.schulbaum03
```

(alles in einer Zeile)

Die Ausgabe sieht dann so aus:

```
Replacing file ownership for path: DOCS:S01/home/lehrer/SpechtB-S01 oldDN:
oldGUID:e1b3b28c-6936-49e8-d1-ad-8cb2b3e13669
Replacing file ownership for path: DOCS:S01/home/lehrer/SperlingH-S01 oldDN:
oldGUID:e1b3b28c-6936-49e8-d1-ad-8cb2b3e13669
Operation Succeeded.
Changed ownership of 2 files/folders
```

Dies können wir mit dem List-Befehl

```
gserver03:~ # nsschown -l DOCS:S01/home/lehrer -a
```

überprüfen:

```
Path::DOCS:S01/home/lehrer| Owner
DN::.CN=admin.OU=Server.OU=DIENSTE.O=m13.T=SCHULBAUM03.| Owner GUID::c47579bc-c8c2-41f3-98-be-bc7975c4c2c8
Path::DOCS:S01/home/lehrer/SpechtB-S01| Owner
DN::.CN=admin.OU=Server.OU=DIENSTE.O=m13.T=SCHULBAUM03.| Owner GUID::c47579bc-c8c2-41f3-98-be-bc7975c4c2c8
Path::DOCS:S01/home/lehrer/SperlingH-S01| Owner
DN::.CN=admin.OU=Server.OU=DIENSTE.O=m13.T=SCHULBAUM03.| Owner GUID::c47579bc-c8c2-41f3-98-be-bc7975c4c2c8
Path::DOCS:S01/home/lehrer/SperlingH-S01/public.html| Owner DN::.CN=SperlingH-S01.OU=Lehrer.OU=Benutzer.OU=S01.OU=SCHULEN.O=m13.T=SCHULBAUM03.| Owner GUID::a6165b6e-8f32-4550-bd-b3-6e5b16a6328f
Path::DOCS:S01/home/lehrer/SpechtB-S01/Test.txt| Owner DN::.CN=SpechtB-S01.OU=Lehrer.OU=Benutzer.OU=S01.OU=SCHULEN.O=m13.T=SCHULBAUM03.| Owner GUID::ea91faae-9257-469d-0e-a9-ae9a91ea5792
```

Operation Succeeded.

Dabei bleiben die Owner der Datei *Test.txt* und des Unterverzeichnisses *public_html* unverändert erhalten, denn diese haben ja eine ganz andere GUID.

2. Allgemeine Benutzer gelöscht

Wollen wir nicht so gezielt vorgehen, wie in Kap. 1, suchen wie einfach alle Dateien/Ordner ohne gültigen Owner und ersetzen den ungültigen Owner durch den *admin*.

Hier ein kleines vereinfachtes Beispiel. Sagen wir, in den Projekten gibt es ein Projektverzeichnis *Mathematik*, in dem der Benutzer *SperlingH* eine Datei *Newton.doc* abgelegt hat.

Schauen wir uns dies mit *nsschown* an:

```
gserver03:~ # nsschown -l DATA:S01/projekte -a
```

Ausgabe:

```
Path::DATA:S01/projekte| Owner DN::.CN=admin.OU=Server.OU=DIENSTE.O=m13.T=SCHULBAUM03.| Owner GUID::c47579bc-c8c2-41f3-98-be-bc7975c4c2c8
Path::DATA:S01/projekte/Mathematik| Owner DN::.CN=BenAdmin-S01.OU=BenAdmins.OU=Verwalter.OU=Benutzer.OU=S01.OU=SCHULEN.O=m13.T=SCHULBAUM03.| Owner GUID::dfeaf798-50ea-4de5-10-9a-98f7eadfea50
Path::DATA:S01/projekte/Mathematik/Newton.doc| Owner DN::.CN=SperlingH.OU=Lehrer.OU=Benutzer.OU=S01.OU=SCHULEN.O=m13.T=SCHULBAUM03.| Owner GUID::a6165b6e-8f32-4550-bd-b3-6e5b16a6328f
```

Operation Succeeded.

(In Ihrer Schule würden hier natürlich viel mehr Zeilen entstehen.)

Der Ordner *projekte* wurde (während der GServer03-Entwicklung) vom *admin* erzeugt, das Projekt *Mathematik* vom *BenAdmin* und die Datei *Newton.doc* von *SperlingH*.

(Natürlich könnte man auch das gesamte *DATA:S01* oder auch *DATA:* scannen.)

Nun wird der Benutzer *SperlingH* gelöscht. Es ergibt sich:

```
gserver03:~ # nsschown -l DATA:S01/projekte -a

Path::DATA:S01/projekte| Owner DN::CN=admin.OU=Server.OU=DIENSTE.O=ml3.T=SCHULBAUM03.|
Owner GUID::c47579bc-c8c2-41f3-98-be-bc7975c4c2c8
Path::DATA:S01/projekte/Mathematik| Owner DN::CN=BenAdmin-
S01.OU=BenAdmins.OU=Verwalter.OU=Benutzer.OU=S01.OU=SCHULEN.O=ml3.T=SCHULBAUM03.| Owner
GUID::dfeaf798-50ea-4de5-10-9a-98f7eadfea50
Path::DATA:S01/projekte/Mathematik/Newton.doc| Owner DN::| Owner GUID::a6165b6e-8f32-
4550-bd-b3-6e5b16a6328f

Operation Succeeded.
```

Noch gezielter ist der folgende Befehl mit dem Parameter *-i*, der nur die „unbestimmten“ Einträge anzeigt:

```
gserver03:~ # nsschown -l DATA:S01/projekte -i

Path::DATA:S01/projekte/Mathematik/Newton.doc| Owner DN::| Owner GUID::a6165b6e-8f32-
4550-bd-b3-6e5b16a6328f

Operation Succeeded.
```

Der Besitzer von Newton.doc ist nun unbestimmt. Man könnte hier für Spezialfälle nun ähnlich wie in Kap. 1.3 selektiv vorgehen. Aber wir wollen nun generell für alles, was einen unbestimmten Benutzer hat, den *admin* als Owner einsetzen. Dies geschieht mit Hilfe des Parameters *-i*:

```
gserver03:~ # nsschown -l DATA:S01/projekte -i -n admin.server.dienste.ml3.schulbaum03

esetting invalid ownerID GUID: a6165b6e-8f32-4550-bd-b3-6e5b16a6328f for path:
DATA:S01/projekte/Mathematik/Newton.doc
Operation Succeeded.
Changed ownership of 1 files/folders
```

(In Ihrer Schule würden hier natürlich viel mehr Zeilen entstehen.)

Schauen wir uns das Ergebnis an:

```
gserver03:~ # nsschown -l DATA:S01/projekte -a

Path::DATA:S01/projekte| Owner DN::CN=admin.OU=Server.OU=DIENSTE.O=ml3.T=SCHULBAUM03.|
Owner GUID::c47579bc-c8c2-41f3-98-be-bc7975c4c2c8
Path::DATA:S01/projekte/Mathematik| Owner DN::CN=BenAdmin-
S01.OU=BenAdmins.OU=Verwalter.OU=Benutzer.OU=S01.OU=SCHULEN.O=ml3.T=SCHULBAUM03.| Owner
GUID::dfeaf798-50ea-4de5-10-9a-98f7eadfea50
Path::DATA:S01/projekte/Mathematik/Newton.doc| Owner
DN::CN=admin.OU=Server.OU=DIENSTE.O=ml3.T=SCHULBAUM03.| Owner GUID::c47579bc-c8c2-41f3-
98-be-bc7975c4c2c8

Operation Succeeded.
```

Newton.doc hat nun wieder einen gültigen Owner.

3. Wann tätig werden ?

Eine schwierige Frage.

Ein Grund tätig zu werden, ist z.B. das Auftreten einer hohen Last auf dem *Filr*. Dies kann man an der *Filr*-Konsole mit dem Kommando *top* überprüfen, oder mit dem Sichten der Datei */var/opt/novell/filr/log/famtd.log*, in der vielleicht viele/sehr viele Fehler protokolliert wurden.

Tätig werden könnten Sie auch, wenn Sie gerade viele Benutzer gelöscht haben, z.B. beim Schuljahreswechsel.

Bemerkung: Haben Sie z.B. mit Hilfe von *WinSCP* (eingeloggt als *root*) Dateien auf eines der NSS-Volumes kopiert, so gibt es **kein** Problem. Der Owner ist dann *gserver03.server.dienste.ml3*, und den gibt es ja.

4. Schluss

Fazit: Noch mal gutgegangen!

Ihre ZEN-Novell.

Anhang A (Filr)

Im Rahmen eines Novell/Microfocus Service Requests wegen des Problems „Dateien ohne Owner“ wurde uns eine Erläuterung im Zusammenhang mit *Filr*, die wir hier im Original-Wortlaut wiedergeben, mitgeteilt:

- When Filr synchronizes a file for the very first time and does not find the owner in the Filr user database, it assigns File Sync Agent as the owner. This should not incur any additional overhead.
- When Filr does subsequent synchronization on a Net Folder and finds that a file which used to have a valid owner no longer has an owner, it then changes the record in Filr and updates it to the File Sync Agent. Prior to Filr 1.1, we did not do this but starting with Filr 1.1 some users found that we were not doing a good job at updating File Owners and we introduced a default setting which would keep file owners up to date on subsequent synchronizations (reference Bug 864496).

For example: In this scenario, suppose 20,000 files were created or synchronized in Filr by students between Jan 1, 2016 and Dec 1, 2016 and all of them had valid owners. Then all those students were deleted from the directory on Jan 1, 2017 which causes missing owners on those files. Next time these Net Folder(s) have a scheduled sync run on them, we will attempt to change their ownership to File Sync Agent. This is because file ownership is meta-data for the file and a subsequent sync should capture that as a change.

There is a configuration setting in Filr which controls this behavior, and by default, we turned it ON starting Filr 1.1. If you would like to change it to OFF so that we no longer try to update owners, I can share that with you. However, please note that turning it OFF will introduce the original problem that some files show outdated owners.

Wer die im letzten Absatz erwähnte Einstellung ausprobieren möchte, müsste in der Datei */opt/novell/filr/apache-tomcat/webapps/ssf/WEB-INF/classes/config/ssf-ext.properties* den Eintrag

```
mirrored.folder.unmapped.creator.causes.reset=false.
```

hinzugefügen. (Vorher */etc/init.d/teaming stop*. Nachher */etc/init.d/teaming start*)

In jedem Fall sollten Sie nachprüfen, ob die Rechte des *proxyuserfilr.server.DIENSTE.ml3* korrekt gesetzt sind. Im NSS-Filesystem muss dieser das S-Recht auf die Volumes haben, die *Filr* benutzt, also auf DOCS und ggf. auch auf DATA. Die eDirectory-Objektrechte sind:

Auf SCHULBAUM03: (ACL) Read
auf gserver03.server.dienste.ml3: (All Attributes Rights) Read

(Siehe hierzu z.B.: <https://www.novell.com/documentation/filr-3/filr-bp-plan-deploy/data/nf-proxy-user-planning.html>)

Landesmedienzentrum Baden-Württemberg (LMZ)
Support Netz
Rotenbergstraße 111
70190 Stuttgart

© Landesmedienzentrum Baden-Württemberg, 2017