

# OpenCSW – Solaris-Pakete für das Rechenzentrum

Dagobert Michelsen  
Baltic Online Computer GmbH  
Kiel

## Schlüsselworte

Solaris, Pakete, OpenCSW, Portierung, Open Source

## Einleitung

Bei dem Einsatz von Solaris als Betriebssystemplattform in Projekten wird oft neben der eigentlichen Enterprise-Applikation zusätzliche freie Software wie etwa Apache oder Nagios benötigt. Diese selbst zu kompilieren und aktuell zu halten ist aufwendig. Eine Reduktion oder Vermeidung dieser Aufwände wäre wünschenswert und würde die Arbeitsabläufe vereinfachen. Durch den Einsatz fertiger Pakete kann dies erreicht werden. Dies gilt insbesondere bei Software mit vielen Abhängigkeiten. OpenCSW ist ein Paketierungsprojekt mit über 3700 Paketen und damit die größte Quelle für vorkompilierte Software für Solaris. Darunter befindet sich unter anderem verschiedene Webserver, freie Datenbanken, Programmiersprachen und beliebte Utilities. Der Schwerpunkt liegt auf Paketen für Solaris 10<sup>1</sup>, ein IPS Repository für Solaris 11 befindet sich im Aufbau. Die existierenden Pakete für Solaris 10 können bis dahin auch unter Solaris 11 genutzt werden. Die Beschreibungen zum Erzeugen der Pakete mit ggf. notwendigen Patches sind frei unter <http://gar.opencsw.org> verfügbar. Weiterhin arbeitet OpenCSW mit 43 Upstream-Projekten (davon 10 GNU-Projekte) zusammen, um die einfache Übersetzbarkeit unter Solaris sicherzustellen<sup>2</sup>.

Das CSW-Prefix für Pakete wurde als erstes durch den Sun-Mitarbeiter Alan DuBoff verwendet. Kurze Zeit später wurde das CSW-Projekt bei Blastwave durch Dennis Clarke und Phil Brown maßgeblich erweitert und wird seit 2008 durch den OpenCSW-Verein weitergeführt. Durch die Offenlegung aller Projektquellen bei OpenCSW und die 20 offiziellen Spiegel ist der Bestand auf absehbare Zeit gesichert. Die Pakete werden mittlerweile von Banken, Telcos und Energieversorgern weltweit eingesetzt.

## Pakete für Solaris

Als Beispiel soll ein Rechner für OpenCSW vorbereitet und das Tool „top“ installiert werden. Sollte der Rechner über einen direkten Internetzugang verfügen, dann kann als root begonnen werden mit dem folgenden Aufruf:

```
pkgadd -d http://get.opencsw.org/now  
PATH=$PATH:/opt/csw/bin  
pkgutil -U -i top
```

und dann mit „top“ das Binary direkt ausgeführt werden. Dabei wird das von OpenCSW bereitgestellte Werkzeug „pkgutil“ installiert, mit welchem Pakete heruntergeladen und installiert werden können. Bei der Installation werden Abhängigkeiten zu anderen Paketen automatisch aufgelöst und diese ebenfalls geladen und installiert. Die Installation der Pakete erfolgt mit dem systemeigenen pkgadd. Als Konsequenz können die Pakete nur durch Root installiert werden, sind aber auch durch jedes andere auf diesen Standardkommandos aufbauende Provisionierungswerkzeug installierbar. Die Abhängigkeiten werden durch sog. „Kataloge“ aufgelöst, die zu jedem Paket eine textuelle

- 1 Für Solaris 8 und 9 gibt es noch historische Repositories, die aber nur noch sporadisch aktualisiert werden.
- 2 Die bei der Portierung gewonnen Erkenntnisse sind verfügbar unter <http://wiki.opencsw.org/porting-faq>

Beschreibung, die Abhängigkeiten des Paketes, seine Größe und eine Prüfsumme enthalten<sup>3</sup>. Ebenfalls wäre es möglich, nicht-OpenCSW Pakete mit pkgutil zu installieren, um so den einfachen Installationsmechanismus mit Abhängigkeiten nutzen zu können.

### Layout

Die Pakete werden unter `/opt/csw` abgelegt, die Konfigurationsdateien befinden sich unter `/etc/opt/csw` und die Bewegungsdaten unter `/var/opt/csw`. Für viele Pakete werden neben den 32 Bit Standardbuilds zusätzlich optimierte Varianten und 64 Bit angeboten.

### Konfiguration

Die Pakete unterstützen soweit möglich sowohl Fullzones als auch Sparsezones. Daher werden zwei Verzeichnisse mit Konfigurationsdateien verwendet, einmal `/opt/csw/etc` für globale Einstellungen und `/etc/opt/csw` für Anpassungen in nichtglobalen Zonen. Da nicht jede Software zwei Konfigurationsorte unterstützt besitzen einige Pakete nur das flexiblere `/etc/opt/csw`.

### Starten von Prozessen

Die Dienste von OpenCSW haben in der Solaris SMF ein „csw“-Prefix in dem FMRI. Ob Daemons automatisch nach der Installation eines Paketes gestartet werden sollen kann in der `/etc/opt/csw/csw.conf` konfiguriert werden:

```
autoenable_daemons=no
```

### Anlegen eines lokalen Spiegels

In der Standardeinstellung lädt pkgutil die Pakete direkt von dem OpenCSW Hauptspiegel aus dem Internet. Da Solaris-Systeme oft nicht direkt mit dem Internet verbunden sind ist die Anlage eines lokalen Spiegels empfehlenswert, von dem aus dann die Pakete geladen werden können. Zum Anlegen des Spiegels ist `rsync` hilfreich, da zwischen den Katalogen Hardlinks eingesetzt werden:

```
rsync -Hrav rsync://mirror.opencsw.org/opencsw <Zielverzeichnis>
```

Danach muss der Ort des Spiegels für pkgutil in der `/etc/opt/csw/pkgutil.conf` entsprechend angepasst werden. Die Pakete können durch pkgutil über jede URL angezogen werden, die durch `wget` verarbeitet werden kann, z.B. HTTP, FTP oder auch lokal über die FILE-Direktive.

```
mirror=file://<Zielverzeichnis>/unstable
```

### Aktualisierung

Die Kataloge werden regelmäßig mit Updates zu Paketen versorgt. Wenn neue Pakete in dem Katalog zur Verfügung stehen, kann das System automatisch die installierten Pakete aktualisieren mit

```
pkgutil -y -U -u
```

### Releases

Die Pakete werden je nach „Reifegrad“ in unterschiedliche Kataloge einsortiert. Die Namensgebung orientiert sich an Debian mit den Zweigen „stable“ – „testing“ – „unstable“. Darüber hinaus gibt es noch Releases mit Namen, die jeweils die Zweige durchlaufen. Das aktuelle Release in „testing“ ist „dublin“ und „unstable“ ist zurzeit „kiel“<sup>4</sup>.

---

3 Die Kataloge sind darüberhinaus mit PGP signiert. Für Details zur Konfiguration siehe <http://www.opencsw.org/security>

4 Für Details und nähere Erläuterungen siehe <http://wiki.opencsw.org/releases-and-staging>

## Bauen von Paketen

Da die Buildbeschreibungen der Pakete frei verfügbar sind können diese auf einfache Weise Nachgebaut werden. Dies erlaubt den Einsatz von OpenCSW in Umgebungen, in denen nur vor Ort gebaute Binaries eingesetzt werden dürfen. Weiterhin ist dadurch die Paketierung von Software möglich, deren Lizenzbedingungen eine Verteilung in Paketform untersagen, jedoch die Einbindung in die OpenCSW Paketverteilung nützlich ist (z.B. der Oracle Instantclient oder das JRE).

Die Pakete von OpenCSW sind mit dem mGAR Buildsystem gebaut. Um ein Paket nachzubauen muss zuerst das System dafür vorbereitet werden. In den nachfolgenden Schritten wird exemplarisch das Nachbauen des Paketes für "tssh" beschrieben:

1. System für OpenCSW vorbereiten  
`pkgadd -d http://get.opencsw.org/now  
PATH=$PATH:/opt/csw/bin`
2. Gegebenenfalls passenden Spiegel wählen
3. mGAR installieren  
`pkgutil -y -i mgar`
4. Compiler installieren (z.B. SolarisStudio 12.3)  
Herunterladen<sup>5</sup> und auspacken, dann installieren mit  
`cd SolarisStudio12.3-solaris-x86-pkg  
./install_patches.sh  
sh solarisstudio.sh --non-interactive`
5. Benutzer zum Bauen der Pakete anlegen  
`mkdir /export/home/cswbuild  
useradd -d /export/home/cswbuild -s /usr/bin/zsh cswbuild  
chown cswbuild /export/home/cswbuild  
su - cswbuild  
PATH=$PATH:/opt/csw/bin  
echo "GARCOMPILER = SOS12U3" >> ~/.garrc`
6. mGAR Umgebung für diesen Benutzer aufsetzen  
`mgar init  
cd opencsw  
mgar up --all  
git config --global user.email dam@opencsw.org  
git config --global user.name "Dagobert Michelsen"  
echo "SPKG_PACKAGER = Dagobert Michelsen" >> ~/.garrc  
echo "SPKG_EMAIL = dam@opencsw.org" >> ~/.garrc`
7. Gewünschtes Paket nachbauen  
`cd tssh/trunk  
mgar package`
8. Das fertige Paket liegt bereit in  
`~/staging/build-<datum>/  
tssh-6.18.01,REV=2012.09.11-SunOS5.10-i386-CSW.pkg.gz`

Eine Liste aller Buildbeschreibungen und die Dokumentation des mGAR Build Systems befindet sich unter <http://gar.opencsw.org>.

---

5 <http://www.oracle.com/technetwork/server-storage/solarisstudio/downloads/index.html>

## Eigene Pakete

Die von OpenCSW bereitgestellten Pakete werden über Makefiles beschrieben. Neben den fertigen Pakete können auch eigene Pakete durch erstellen von Makefiles erzeugt werden. Die Paketbeschreibungen bestehen im wesentlichen aus Variablendefinitionen. Die Struktur der Makefiles abstrahiert von dem eigentlichen Bauen der Pakete, indem dessen notwendigen Bestandteile wie der Tarball mit den Quellen, der Version und eine kurze Beschreibung definiert werden. Das mGAR Buildsystem führt dann alle notwendigen Schritte entsprechend dieser Definitionen zum Bauen des Paketes aus. Für ein einfaches Paket mit “configure” sieht ein minimales Makefile wie folgt aus:

```
NAME = gperf
VERSION = 3.0.4
CATEGORIES = devel

DESCRIPTION = A perfect hash function generator

MASTER_SITES = $(GNU_MIRROR)
DISTFILES = $(DISTNAME).tar.gz

include gar/category.mk
```

Als erstes muss der tarball geladen werden und seine Prüfsumme berechnet werden. Dies wird in einer separaten Datei zusammen mit dem Makefile gespeichert und stellt zur späteren Wiederverwendung identische Ergebnisse sicher. Dann kann die Software kompiliert und das Paket gebaut werden:

```
mgar makesum
mgar package
```

Die Stärke von mGAR ist die einfache Einbindung von komplexen Features. So ist etwa eine zusätzliche Version mit 64 Bit erzeugbar:

```
BUILD64 = 1
```

Diese kann dann explizit mit `/opt/csw/bin/{sparcv9|amd64}/` angesprochen werden. Soll ISAEXEC verwendet werden ist dies ebenfalls durch eine einzeilige Direktive möglich:

```
ISAEXEC = 1
```

Konfigurationsdateien können automatisch zwischen Updates gesichert werden:

```
PRESERVECONF = /etc/opt/csw/foo.conf
```

Es existieren noch viele weitere Direktiven wie die Einbindung von Diensten in SMF und `init.d`, Ergänzen der `/etc/services`, oder das Hinzufügen von Benutzern<sup>6</sup>.

---

<sup>6</sup> Eine vollständige Liste der sog. Class Action Scripts ist auf <http://wiki.opencsw.org/class-action-scripts> und in der GAR Dokumentation unter <http://gar.opencsw.org> verfügbar

## Community

OpenCSW lebt vom Mitmachen und die besten Maintainer sind solche, die ihre Pakete selbst benutzen. Sollten sie ein Paket bei OpenCSW vermissen oder würden gerne die Paketierung verbessern ist die einfachste Möglichkeit einfach dem Projekt beizutreten und so die Erweiterungen einzubringen. Zweimal im Jahr finden Treffen von Maintainern und interessierten Benutzern in Europa statt<sup>7</sup> (u.a. München, Zürich, Oslo, Dublin, Bratislava) um das System konstruktiv und organisatorisch voranzutreiben. Das nächste Camp findet voraussichtlich vom 9. bis 11. November in Lenggries-Fall bei München statt.

*Es ist eine Session zum Bauen von Paketen im Rahmen der Unconference geplant!*

## Wichtige Kontakte zu dem Projekt

Bei Fragen, Anregungen oder Hilfe bei der Erstellung von Paketen gibt es verschiedene Kontaktmöglichkeiten:

<http://www.opencsw.org/community>

<http://www.opencsw.org/bugs>

<http://freenode.net> IRC-Channel #opencsw

Twitter: #opencsw

## Kontaktadresse:

Dagobert Michelsen  
Baltic Online Computer GmbH  
Alter Markt 1-2  
D-24103 Kiel

Telefon: +49 (0) 431-54003 0

Fax: +49 (0) 431-54003 99

Mobil: +49 (0) 170-445 7772

E-Mail: [dam@baltic-online.de](mailto:dam@baltic-online.de)

Internet: <http://www.baltic-online.de>

[dam@opencsw.org](mailto:dam@opencsw.org)

<http://www.opencsw.org>

---

<sup>7</sup> Für Details und Fotos siehe <http://wiki.opencsw.org/camps>