

Die Schnittstelle zwischen SAP DB und Perl

STEFAN WICHMANN, ADMINISTRATOR UND ENTWICKLER

SAP DB ist die frei verfügbare Adabas-D-Variante der SAP AG. Natürlich harmoniert SAP R/3 vorzüglich mit dieser Datenbank, wobei SAP DB aber auch standalone verwendet werden kann. Nachdem bereits in der Ausgabe 1'02 der freeX die Installation dieses Systems beschrieben wurde, geht es diesmal um die Ansprache der Datenbank aus externen Programmen. Dafür bietet sich die Schnittstelle zu Perl an.

Im folgenden wird gezeigt, wie die Datenbank aus externen Programmen heraus angesprochen wird. Man könnte die an dieser Stelle entwickelte Erweiterung der Datenbank in das Programm zur Loganalyse einbauen, das in der Ausgabe 2'2003 der freeX vorgestellt wurde [2]. Aus diesem Grund geht es jetzt um die Anbindung mit der Programmiersprache Perl.

Die Installation des Perl-Interpreters erfolgt bei vielen Linux-Distributio-

nen bereits im Standardpaket. Den notwendigen Test, ob Perl installiert und aktiv ist, nimmt man in einem Konsolenfenster mit dem Aufruf `perl -v` vor. Dabei wird die Versionsnummer des Interpreters ausgegeben. Als Grundlage für die Arbeiten dient die mit SAP DB ausgelieferte Beispieldatenbank mit dem Namen »TST«. Sie verfügt über die Benutzer `dba`, `dbm` und `test`.

SAP DB besteht wie jede SQL-Datenbank aus einem Server, der im

Hintergrund gestartet wird und dann auf Anfragen der Clients wartet. Den Server fährt man mit dem Auf von `x_server` hoch. Bei der Standard-Installation befindet er sich in `»/opt/sapdb/indep_prog/bin«`. Ist dieses Verzeichnis nicht in die Pfad-Variablen aufgenommen, muß der komplette Pfad beim Aufruf mit angegeben werden. Man darf sich nicht davon irritieren lassen, daß der Server die Erfolgsmeldung »Vserver startet« ausgibt.

```
#!/usr/bin/env perl
use sapdb;
use demodbData;

sub sayHello(session) {
    my($session) = @_;
    my($cursor) = $session->sql(
        "CREATE TABLE table_name(nachname CHAR(12))");
    my($cursor) = $session->sql(
        "INSERT INTO table_name(nachname) VALUES('wichmann')");
    my($cursor) = $session->sql(
        "SELECT COUNT(*) FROM table_name");
    my($cursor) = $session->sql(
        "SELECT NAME FROM table_name");
    my($row) = $cursor->next();
    my($result);
    # row is a tuple with one element
    ($result,) = $row->[0];
    print "$result\n";
}

sub createSession {
    my($username, $password, $dbname, $host) = @_;
    my($session) = sapdb::connect($username, $password,
        $dbname, $host);
    return $session;
}

sub main {
    my(@argv) = @ARGV;
    my($username, $password, $dbname,
        $host) = parseUserArgs(@ARGV);
    my($session) = &createSession($username, $password,
        $dbname, $host);
    &sayHello($session);
    $session->commit(); # not necessary for a Select,
        # but good style anyway
}
}
```

Listing 1: Das Perl-Skript zur Verwaltung der Beispieldatenbank [3]

Damit ist die Grundvoraussetzung für den Start des Clients geschaffen. Er wird mit dem Aufruf

```
dbmcli -s -d TST -u dbm,dbm db_start
```

gestartet. Die Datenbank ist jetzt zwar gestartet, jedoch noch nicht bereit, Befehle entgegenzunehmen. Sie befindet sich im »kalten« Zustand (»cold«). Um die Datenbank nutzen zu können, ist zusätzlich die folgende Initialisierung nötig:

```
dbmcli -s -d TST -u dbm,dbm db_warm
```

Der Status der gestarteten Datenbank wird mit dem Befehl

```
dbmcli -s -d TST -u dbm,dbm db_warm
```

abgefragt, so daß sichergestellt ist, daß der Datenbankserver wirklich läuft.

Zur Nutzung von SAP DB eignet sich Perl. Es müssen in Perl zwei Module für datenbankspezifische Befehle geladen werden. Nach dem Einbinden von *sapdb* mit *use* steht der neue Befehl *sql* zur Verfügung. Ihm übergibt man dann die üblichen SQL-Statements wie *Create Table* oder *Select* als Parameter (Listing 1). Das Packagemodul mit dem Namen »demodbData.pm« enthält die Benutzer, die in der Datenbank »TST« bekannt sind. Listing 2 zeigt das Programm, mit dem eine Tabelle gelöscht wird.

Auf diese Weise können aus Perl-Programmen beliebige SQL-Anweisungen an den Datenbank-Server gesandt werden. Eine andere Möglichkeit besteht im direkten Aufruf

des bereits angesprochenen Kommandozeilen-Clients *dbmcli*, der sich nach der Installation in »/opt/sapdb/indep_prog/bin« befindet und den man normalerweise über den vollen Pfadnamen aufruft. Alternativ empfiehlt es sich, dieses Verzeichnis in den Pfad aufzunehmen.

```
sapdb@linux:> dbmcli -d TST -u dbm,dbm
```

Sobald das Tool aktiv ist, erfolgt die Anmeldung an die Datenbank über die Anweisung *sql_connect*, was über eine OK-Meldung quittiert wird. Ab sofort können auch so beliebige SQL-Befehle abgesetzt werden. ◆

Literatur:

- [1] Matthias Süß: SAP DB unter Linux installieren, freeX 1'2002, S. 26 ff.
- [2] Stefan Wichmann: Logs in SAP R/3, freeX 2'2003, S. 40 ff.
- [3] Modifiziert nach HelloSAPDB.pl

```
#!/usr/bin/env perl
use sapdb;
use demodbData;

sub sayHello(session) {
    my($session) = @_;
    my($cursor) = $session->sql("DROP TABLE table_names1");
}

sub createSession {
    my($username, $password, $dbname, $host) = @_;
    my($session) = sapdb::connect($username, $password,
                                  $dbname, $host);
    return $session;
}

sub main {
    my(@argv) = @ARGV;
    my($username, $password, $dbname,
        $host) = parseUserArgs(@ARGV);
    my($session) = &createSession($username, $password,
                                   $dbname, $host);

    &sayHello($session);
    $session->commit();
}

&main();
```

Listing 2: Löschen der Tabelle

freeX

Autoren gesucht

Kennen Sie sich mit freien Unix-Systemen aus? Können Sie Ihr Wissen an andere Leser vermitteln? Haben Sie Interesse an einem attraktiven Honorar? Dann schreiben Sie doch für die *freeX*! Wir suchen immer kompetente Autoren, die ihr Wissen weitergeben wollen. Ob Sie pfiffige Tips und Tricks auf Lager haben oder Grundlagen vermitteln wollen, Sie sind auf jeden Fall bei uns richtig! Senden Sie uns doch einen kurzen Abriß über das Thema, über das Sie schreiben wollen, und wir setzen uns postwendend mit Ihnen in Verbindung.

Unsere E-Mail-Adresse hierfür lautet: freex@cul.de Selbstverständlich können Sie uns auch dem Postweg erreichen. – Adressieren Sie Ihren Brief an:
 Redaktion *freeX*
 10 Rue des Hauts Champs
 F-88110 Luvigny
 Tel. (0033) 3.29.42.40.02
 Fax (0033) 3.29.42.40.03