Conversion Tools - Feature #5819

update standard project build.xml to provide targets for backup/restore

11/11/2021 03:07 PM - Greg Shah

Status:	New	**Start date:**	
Priority:	Normal	**Due date:**	
Assignee:		**% Done:**	0%
Category:		**Estimated time:**	0.00 hour
Target version:			
billable:	No	**version:**	
vendor_id:	GCD		

Description

History

#1 - 11/11/2021 03:10 PM - Greg Shah

From Eric:

> I know it takes a very long time to import large databases, like <large_customer_application>. This tends to become a blocker whenever we need to refresh the database.
>
> If you have the available space, please note that restoring a working database from a template/master database (a "live backup") is a faster alternative. Creating and restoring from a live backup is described in the documentation here: Live Backup.
>
> Please note that you still should install the latest p2jpl.jar and run the analyze command after the restore of the live backup to a working database. The analyze step still can take quite a bit of time, though the database can be used while this is running (some queries may be slow, however, until analyze has finished updating its statistics).
>
> I have an SSD, where I usually have my working databases, and a slower drive, where I keep my master, live backups. PostgreSQL allows you to create different tablespaces for exactly this purpose (see the documentation for CREATE TABLESPACE in PostgreSQL docs). Given the same media speed, restoring from a live backup always has been faster for me than a full import from *.d files. On my system, it took maybe an hour to restore the previous <large_customer_application> database from my slow drive to my SSD. I haven't measured the time for the most recent drop; it is bigger, so I'm sure it will take longer than that. YMMV.
>
> Of course, the drawback of this approach is that many of the databases we use are quite large, and you may not have the space available to maintain live backups. In that case, the alternative is unfortunately, to re-run the import.
>
> As a reference, I'm attaching the script I use to restore the <large_customer_application> database from a live backup. I run it from the parent directory of the <large_customer_application> project. It is pretty specific to my system (my cluster port number, tablespace names, etc.), but it can be adapted accordingly. It assumes the live backup has the same name as the working database, plus the suffix "_master". We may want to integrate something like this into a build target eventually, if we want this to be an option from ant.

```
# Restore pristine working database from template

if [ $# != 2 ]
then
   arg1="<working database name>"
   arg2="<path to project root>"
   echo "Use: $0 $arg1 $arg2"
   echo "     $arg1 will be dropped and recreated from template ${arg1}_master"
   echo "     p2jpl.jar will be installed using ${arg2}/p2j/pl/install_p2j.sql"
   exit
fi

port=5434

admin=fwd_admin
user=fwd_user
db=$1
loc=$2
space=pg_default
#space=slow

export PGPASSWORD=admin
```

```
# drop old working database
dropdb -U $admin -p $port -h localhost $db

# create copy of template database owned by non-superuser
createdb -U $admin -O $user -p $port -h localhost -T ${db}"_master" -D $space $db

# install base PL/Java support
psql -U $admin -p $port -h localhost -c "create extension pljava" $db

# install P2J PL/Java UDFs
psql -U $admin -p $port -h localhost -f ${loc}"/p2j/pl/install_p2j.sql" $db

# grant permission to use PL/Java schema
psql -U $admin -p $port -h localhost -c "grant usage on schema sqlj to public" $db

# update statistics
psql -U $admin -p $port -h localhost -c analyze $db
```

#2 - 11/11/2021 03:10 PM - Greg Shah

We should have build targets for the backup that creates the master and the restore from the master.