refreshdb User Guide

Author: Ian Jones, IMiJ Ltd
Version: 2.1
Issued Date: 15th August 2004

Contents

refreshdb User Guide....1
Contents....2
Introduction....3
Configuration And Pre-Requisites....4
Main Configuration File....4
Optional Section....4
SOURCE_DB....4
TARGET_DB....4
INCLUDE_FILE....5
EXCLUDE_FILE....5
DELETE_FILE....5
REFSQL_FILE....5
Required Section....5
TMP_DIR....5
UNL_DIR....6
MAXPROC_UNLOAD....6
MAXPROC_LOAD....6
MAXPROC_SLEEP....6
MAXPROC_SLEEP_CHECK....6
MAXREC....6
Include File....7
Exclude File....7
Delete File....7
Referential SQL File....7
Running refreshdb....9
Command Line Options....9
-k....9
-l....9
-u....9
-d delete_file....9
-e exclude_file....9
-i include_file....9
-r refsql_file....9
-s sourcedb@server....9
-t targetdb@server....9
Process Flow....10
Version History....11
Contact Details....12

Introduction

refreshdb is a UNIX Bourne shell script that can be used when needing to copy data from one Informix database to another without wanting to use a time-consuming backup and restore process or the high performance loader.

The refreshdb script allows for a sub-set of the source database's data to be retrieved and loaded into the target database while maintaining full referential integrity by specifying the data selection SQL. Tables that are to be fully refreshed from the source database do not need to have any SQL specified.

refreshdb may also be used to only partially refresh a target database by specifying which tables to include or exclude from the process, tables may just be cleared out on the target too.

All configuration parameters can be held in simple configuration files, some may be specified on the command line.

The script does not require any user intervention once started so may be used quite happily in a batch process.

Configuration And Pre-Requisites

When using refreshdb you must have the INFORMIXSQLHOSTS environment variable set to a sqlhosts file that contains entries for both the source and target database servers. If however you simply require to copy data between two databases on the same server, this is not a problem either.

The userid running refreshdb only needs to have select permissions on the source database (it can even be the secondary on a Data Replication pair), but must have DBA privileges on the target database.

refreshdb may use up to 5 separate configuration files, each having a distinct purpose. The following sections detail each of the configuration files, starting with the only required configuration file.

Main Configuration File

This configuration file must always be used when running refreshdb. The file may be called anything you like if you use the –f command line option (see: Running refreshdb), the default is "refreshdb.conf". If you were to rename the refreshdb script itself then the default would be "newname.conf" where "newname" is what the refreshdb script has been renamed to.

Each of the variables held in the main configuration script are set using standard Bourne shell syntax, i.e VARIABLE="value".

The main configuration file has two main sections, optional and required.

Optional Section

This contains variables that may be over-ridden with command line arguments to refreshdb or may be entirely omitted, the variables are described below:

SOURCE_DB

SOURCE_DB should be set to the database that contains the data you wish to copy, the "source database". The database may be specified by it's name alone, but most likely should be named with it's server.

```
SOURCE_DB="mydb@server1"
```

If you do not specify a server then you can only copy data from one database to another on the same server. It is highly recommended that you always specify the server name to ensure you have the correct source database.

The SOURCE_DB variable does not need to be assigned if the –s option is used on the command line (see: Running refreshdb). If the –s option is used, it will over-ride the SOURCE_DB entry in the main config file.

TARGET_DB

TARGET_DB should be set to the database that you wish to copy data to, the "target database". The database may be specified by it's name alone, but most likely should be named with it's server.

```
SOURCE_DB="mydb@server2"
```

If you do not specify a server then you can only copy data from one database to another on the same server. It is highly recommended that you always specify the server name to ensure you have the correct target database.

The TARGET_DB variable does not need to be assigned if the –t option is used on the command line (see: Running refreshdb). If the –t option is used, it will over-ride the TARGET_DB entry in the main config file.

INCLUDE_FILE

INCLUDE_FILE should be set the filename of the configuration file that contains a list of tables to be included in the refresh process (see: Include File).

This option is entirely optional, and may be over-ridden by use of the -i option on the command line (see: Running refreshdb).

EXCLUDE_FILE

EXCLUDE_FILE should be set to the filename of the configuration file that contains a list of tables that should not be included in the refresh process (see: Exclude File).

This option is entirely optional, and may be over-ridden by use of the -e option on the command line (see: Running refreshdb).

DELETE_FILE

DELETE_FILE should be set to the filename of the configuration file that contains a list of tables that should have all their records deleted during the refresh process (see: Delete File).

This option is entirely optional, and may be over-ridden by use of the -d option on the command line (see: Running refreshdb).

REFSQL_FILE

REFSQL_FILE should be set to the filename of the configuration file that contains a list of tables and corresponding SQL for selecting a sub-set of data from the source to be copied to the target during the refresh process (see: Referential SQL File).

This option is entirely optional, and may be over-ridden by use of the -r option on the command line (see: Running refreshdb).

Required Section

The following settings in the main configuration file are mandatory, and can only be set in the config file. Once these settings have been set for a particular system, it is very unlikely that you will need to change them again.

TMP_DIR

The TMP_DIR variable holds the name of a directory in which refreshdb can store certain temporary files while it is running.

This directory will be removed (if it exists) and re-created on each invocation of the script. For this reason it is best to leave it set as a local hidden directory name that does not already exist, the default value is:

```
TMP_DIR="./.refreshdb.tmp.d"
```

UNL_DIR

The UNL_DIR variable should point to a existing directory under which refreshdb may store the unloaded data from the source database.

This directory needs to be on a disk partition/slice that has plenty of free space, otherwise you ain't going to be able unload all your data!

This setting **must** be changed before running the script for the first time.

refreshdb creates sub-directories under the specified directory, each directory is named after the source database, so you may find a "mydb@server1" directory or similar.

MAXPROC_UNLOAD

MAXPROC_UNLOAD denotes the number of tables that can be processed during the unload stage at any one time.

Refreshdb processes MAXPROC_UNLOAD number of tables in parallel for the unloads. You may change this if you find that it is using too much resource on your system, or if you think it could use more.

MAXPROC_LOAD

MAXPROC_LOAD denotes the number of tables that can be processed during the load stage at any one time.

Refreshdb processes MAXPROC_LOAD number of tables in parallel for the loads. You may change this if you find that it is using too much resource on your system, or if you think it could use more.

Setting this variable high could produce problems when loading data in the target, if you encounter any locking or long transaction problems try lowering this setting.

MAXPROC_SLEEP

MAXPROC_SLEEP sets the number of seconds to wait before starting the next parallel process, and is purely there to help reduce system and network load to the database servers. A setting of 1 second is generally enough, but you should experiment.

MAXPROC_SLEEP_CHECK

MAXPROC_SLEEP_CHECK is used to set the number of seconds to wait after hitting the MAXPROC level before checking to see if it possible to start any new processes. A setting of 3 to 5 seconds generally seems to be OK, but you should experiment.

MAXREC

When loading data into the target database, MAXREC is used to determine how many records should be inserted per table in a single transaction. This setting is needed to make sure that the transaction volume limits of your database server are not violated, doing so may result in data not being loaded as long transaction errors will be raised.

You may find that you need to tweak this setting or your transaction limits if unloading tables with a very large number of records. However, if you are throwing around multi-millions of records, you probably shouldn't be using this script anyway!

Include File

The include configuration file is optional.

In the include configuration file you may list all the target database tables whose data should be refreshed from the source database. By using the include file you are restricting the refresh to only those tables identified, no other tables will be refreshed. If you do not specify a include file, all tables in the target database that have a companion in the source database would be completely refreshed. For example:

Source db has tables taba, tabb, tabc and tabd
Target db has tables taba, tabb, tabc and tabe

With no include file, the tables taba, tabb, and tabc would be refreshed in the target db, tabe can not be refreshed because the table does not also exist in both the source and target database (see: Referential SQL File for ways of refreshing differently named or differently columned tables).

If a include file containing taba, tabc were to be used, only tables taba and tabc would be refreshed, tables tabb and tabe on the target would not be touched.

Each table name should be entered on a separate line of the file.

Exclude File

The exclude configuration file is optional.

In the exclude configuration file you may list all the target database tables whose data should **not** be refreshed from the source database. By using the exclude file you are restricting the refresh to only those tables which have not been identified in the list.

If the include file has been used then only tables in the include list and not in the exclude list will be refreshed, otherwise all tables not in the exclude file that exist in both the source and target databases will be refreshed.

Each table name should be entered on a separate line of the file.

Delete File

The delete configuration file is optional.

In the delete configuration file you may list all the target database tables whose data should be deleted.

Although you can list any number of tables in the delete file, if they are not in the "refresh list" after processing the include and / or exclude files, they will not be processed. For example, if the target database has three tables, taba, tabb and tabc, with no include file, but a exclude file containing tabb, and the delete file contains tabb and tabc, taba would be refreshed, tabb would not be touched at all and tabc would be cleared of all data.

Each table name should be entered on a separate line of the file.

Referential SQL File

The referential sql (refsql) file is optional.

The refsql file contains a list of tables and related sql that should be used to unload the required data from the source database.

The file has a pipe (|) delimited format, the first field is a table name, the second field is a quoted sql fragment to be used for selecting the tables data.

By using this file you can limit the data retrieved from the source database to a sub-set of data, e.g. a specific date or numeric range. You can use any bonified select statement that can be run against an Informix database, including union joined selects and selects containing case statements or stored procedures. For example:

Target table taba has columns a int, b char(10), c date

You only want two weeks worth of data to be refreshed, a week either side of today's date, so the refsql entry for taba could be:

```
taba|"select * from taba where c >= today - 7 and c <= today + 7"
```

Another use of the refsql file is to select data from a source that does not match the target, for example, you could select columns from any number of joined tables that match the column definition of the target database. For example:

Target table taba has columns a int, b char(10), c date
Source table tabb has columns a int, b char(10)
Source table tabc has columns a int, c date

Tables tabb and tabc have a 1 to 1 relationship using the integer column a.

The refsql entry for taba could be:

```
taba|"select b.a, b.b, c.c from tabb b, tabc c where tabb.a = tabc.a"
```


Running refreshdb

refreshdb may be configured by editing a few simple config files, or by specifying options on the command line. If you type "refreshdb –help" you should see the following:

```
Usage: ./refreshdb [ -klu ] [ -f conf_file ]
 [ -d delete_file ] [ -e exclude_file ] [ -i include_file ] [ -r refsql_file ]
 [ -s sourcedb@server] [ -t targetdb@server ]
```

Command Line Options

The following command line options may be used with refreshdb:

-k

This option specifies that the temporary directory that is created while processing should not be deleted when refreshdb finishes successfully, it should be kept.

-l

This option specifies that the script should not load any data into the target database, the tables that should have been loaded into will however have all their records removed.

-u

This option specifies that the script should not unload any data from the source database if a unload file already exists in the unload directory that does not have zero length.

-d delete_file

You may specify the delete configuration file with this option, this will over-ride any entry made in the main configuration file (see: Main Configuration File and Delete File).

-e exclude_file

You may specify the exclude configuration file with this option, this will over-ride any entry made in the main configuration file (see: Main Configuration File and Exclude File).

-i include_file

You may specify the include configuration file with this option, this will over-ride any entry made in the main configuration file (see: Main Configuration File and Include File).

-r refsql_file

You may specify the refsql configuration file with this option, this will over-ride any entry made in the main configuration file (see: Main Configuration File and Referential SQL File).

-s sourcedb@server

You may specify the source database with this option, this will over-ride any entry made in the main configuration file (see: Main Configuration File).

-t targetdb@server

You may specify the target database with this option, this will over-ride any entry made in the main configuration file (see: Main Configuration File).

Process Flow

The following details in what order refreshdb processes it's tasks once started.

1. Reads and validates command line options and configuration files.
2. Re-creates directory for storing temporary files.
3. If unload directory for source database does no exist, creates it.
4. Creates a "refresh list" of tables common to both target and source.
5. If include file to be used, alters refresh list to only include tables common to both databases and include list.
6. If exclude file to be used, alters refresh list to only include tables common to both databases, include list and that are not in exclude list.
7. If delete file to be used, removes any tables from delete list that do not appear in refresh list, removes delete list tables from refresh list.
8. Data unloaded from source database for every table in the refresh list:
    9. If the -u option has been used, only unload if no unload file already exists or the unload file has zero length.
    10. If any table has an entry in the refsql file, the related sql will be used to select the source data instead of just selecting all records.
    11. The data to be loaded is split into smaller chunks to reduce transaction lengths.
12. If there is a delete list, then all tables in the list will be dropped and re-created on the target database one after the other.
13. Tables to be refreshed will also be dropped and re-created on the target database one after the other.
14. If the -l option has not been used, data unloaded from source database will be loaded from each split file into target database for each table in refresh list:
    15. Before loading the data into each table, all constraints, triggers and indexes on the table will be disabled.
    16. Data from the each split file is loaded into the table one after the other.
    17. After loading data into a table, all constraints, triggers and indexes on the table will be re-enabled, followed by an update of low and high statistics for that table.

Version History

REL-2.1 – 15/08/2004:

- Bug Fix: Test of whether to show MAXPROC_* message failed due to == in test condition on some platforms. Should have always been single =.

REL-2.0 – 31/10/2003:

- Implemented MAXPROC_UNLOAD and MAXPROC_LOAD as replacement of MAXPROC parameter to allow a differing number of parallel processes for the unload and load stages. This was implemented because it was found that in most cases there are less issues with multiple unloads than loads, more could be done at once. Splitting MAXPROC into two parameters allowed for this tuning.
- Only show one "MAXPROC* level (num) hit …" message per wait, instead of wasting log space by repeating the same message unnecessarily.

REL-1.6 – 10/09/2003:

- Fixed daft "feature" that enabled triggers, indexes and constraints after each chunk of data was loaded. Now enables after all chunks loaded. Seriously improves the speed of the script!
- Various small updates to supporting files and added contact details.

REL-1.5 – 08/09/2003:

- Fixed bug where one more process ran than requested.
- Removed -a from split again (oops).

REL-1.4 – 25/08/2003:

- Fixed bug where REFSQL file always used "refreshdb.ref" for file.
- Changed the split of the unload file to happen on unload rather than load to allow more than one load after initial unload complete.
- Uses lock mode of wait and committed read for isolation level.
- Updated and added a number of comments to source.
- Updated example configuration files with better and more generic example data.

REL-1.3 – 01/07/2002:

- Initial public release.

Contact Details

You can contact me (Ian Jones, the author of refreshdb) by email at ian@ianmjones.net

Alternatively, visit my website at http://www.ianmjones.net/ to get the latest version of this package or to find other means of contacting me.

Any comments or suggestions about this package are more than welcome.

Hope you find it useful!