RDSv3 to RDS 2.XX Format Text Files

February 7, 2023

The NSRL has received many inquiries about the further publication of RDS 2.XX format text files. The NSRL will no longer be publishing the RDS data set in this format. However, by following these instructions, an RDS 2.XX format set of text files, and UDF image, can be created using the RDSv3 minimal database publications.

The instructions in this document are intended for a Unix based command line application.

Initial Steps

The first step will be downloading a full minimal database publication from the NSRL RDS downloads web page, found here: https://www.nist.gov/itl/ssd/software-quality-group/national-software-reference-library-nsrl/nsrl-download/current-rds

The database should be updated with the latest delta set publications, for the database set downloaded, also found on the NSRL RDS downloads web page.

Once the database is up to date with the latest published data, create a backup of the database. These instructions will use the backup of the database, called backup.db in this document, in order to create the RDS 2.XX format text files.

```
cp <minimal database>.db backup.db
```

NSRLFile.txt

```
sqlite3 backup.db
DROP TABLE IF EXISTS EXPORT;
CREATE TABLE EXPORT AS SELECT sha1, md5, crc32, file_name, file_size, package_id FROM FILE;
UPDATE EXPORT SET file_name = REPLACE(file_name, '"', '');
.mode csv
.headers off
.output output.txt
SELECT '"' || sha1 || '"', '"' || md5 || '"', '"' || crc32 || '"', '"' || file_name || '"', file_size,
package_id, '"' || 0 || '"', '"' || '"' FROM EXPORT ORDER BY sha1;
.q
sed -i 's/"""/"/g' output.txt
```

Create a header file with a single line containing the following, called NSRLFile-header.txt:
"SHA-1","MD5","CRC32","FileName","FileSize","ProductCode","OpSystemCode","SpecialCode"

```
cat output.txt >> NSRLFile-header.txt
sed -e "s/\r//g" NSRLFile-header.txt > NSRLFile.txt
rm output.txt NSRLFile-header.txt
```

This should give a complete and correct NSRLFile.txt

NSRLMfg.txt

```
sqlite3 backup.db
DROP TABLE IF EXISTS EXPORT;
CREATE TABLE EXPORT AS SELECT manufacturer_id, name FROM MFG;
UPDATE EXPORT SET name = REPLACE(name, '"', '');
.mode csv
.headers off
.output output.txt
SELECT manufacturer_id, '"' || name || '"' FROM EXPORT ORDER BY manufacturer_id;
.q
sed -i 's/"""/"/g' output.txt
```

Create a header file with a single line containing the following, called NSRLMfg-header.txt:
"MfgCode","MfgName"

```
cat output.txt >> NSRLMfg-header.txt
sed -e "s/\r//g" NSRLMfg-header.txt > NSRLMfg.txt
rm output.txt NSRLMfg-header.txt
```

This should give a complete and correct NSRLMfg.txt

NSRLOS.txt

```
sqlite3 backup.db
DROP TABLE IF EXISTS EXPORT;
CREATE TABLE EXPORT AS SELECT operating_system_id, name, version, manufacturer_id FROM OS;
UPDATE EXPORT SET name = REPLACE(name, '"', ''), version = REPLACE(version, '"', '');
.mode csv
.headers off
.output output.txt
SELECT operating_system_id, '"' || name || '"', '"' || version || '"', manufacturer_id FROM EXPORT ORDER BY
operating_system_id;
.q
sed -i 's/"""/"/g' output.txt
```

Create a header file with a single line containing the following, called NSRLOS-header.txt:
"OpSystemCode","OpSystemName","OpSystemVersion","MfgCode"

```
cat output.txt >> NSRLOS-header.txt
sed -e "s/\r//g" NSRLOS-header.txt > NSRLOS.txt
rm output.txt NSRLOS-header.txt
```

This should give a complete and correct NSRLOS.txt’

NSRLProd.txt

```
sqlite3 backup.db
DROP TABLE IF EXISTS EXPORT;
CREATE TABLE EXPORT AS SELECT package_id, name, version, operating_system_id, manufacturer_id, language,
application_type FROM PKG;
UPDATE EXPORT SET name = REPLACE(name, '"', ''), version = REPLACE(version, '"', '');
.mode csv
.headers off
.output output.txt
SELECT package_id, '"' || name || '"', '"' || version || '"', operating_system_id, manufacturer_id, '"' ||
language || '"', '"' || application_type || '"' FROM EXPORT ORDER BY package_id;
.q
sed -i 's/"""/"/g' output.txt
```

Create a header file with a single line containing the following, called NSRLProd-header.txt:
"ProductCode","ProductName","ProductVersion","OpSystemCode","MfgCode","Language","ApplicationType"

```
cat output.txt >> NSRLProd-header.txt
sed -e "s/\r//g" NSRLProd-header.txt > NSRLProd.txt
rm output.txt NSRLProd-header.txt
```

This should give a complete and correct NSRLProd.txt

Creating an RDS 2.XX Format UDF Image

If you need the files in the RDS 2.XX ISO format (a UDF image) for ingest, the following steps will complete the ISO creation.

```
mkdir RDS_2.<rds 3 version>
cp NSRL*txt RDS_2.<rds 3 version>
cd RDS_2.<rds 3 version>
```

A readme file and a signatures file is required in an RDS 2.XX format publication, so they must be manually added.
NSRLFile.txt must also be zipped, and have a NSRLFile.txt.hash file created.

Create a read_me.txt file, including the following text:
This version of the RDS was created manually using an RDSv3 publication.
This is not an official publication of the National Software Reference Library.

Zip NSRLFile.txt and create the NSRLFile.txt.hash file:

```
zip -9 --junk-paths NSRLFile.txt.zip NSRLFile.txt
rm NSRLFile.txt
```

Create a file, NSRLFile.txt.hash, and add the output of the following two commands to the file:

```
openssl sha1 NSRLFile.txt
openssl md5 NSRLFile.txt
```

Create a signatures.txt file, using the following command:

```
openssl sha1 * > signatures.txt
```

Finally, create the UDF image ISO file.

```
genisoimage -udf -allow-limited-size -input-charset=utf-8 -V RDS_2.<rds 3 version> -o "<full path to
current directory>/RDS_<rds 3 version>.iso" <full path to>/RDS_<rds 3 version>
```

For further questions, please contact the NSRL team at nsrl@nist.gov .