IBM Network Station Family of Thin Clients

Access for today, flexibility for tomorrow

Network Station Manager V2R1 Command Line Utility

September 6, 1999

e-business

IBM

Command Line Utility
What is it ?

- An interface to IBM Network Station configuration
- A separate application from Network Station Manager
- The Command Line Utility is used for making batch changes to IBM Network Station XML configuration files
- Quantities of changes may be made quickly
- The same set of changes may be rerun

IBM

Command Line Utility
Who uses it ?

- Used by administrators only !
- Not for individual Network Station users
- User profile that starts Command Line must have;
  - Authority to read and write all files in .../profiles/...
  - Authority to change file access authorities in .../profiles/...
- AS/400 *SECADM and *ALLOBJ special authority required
- Windows NT administrator authority required (use NSMAdmin)
- AIX root or read, write, delete to all files in/below profiles dir

IBM

Command Line Utility
Where does it run ?

- Runs on servers or clients
- Java application that requires Java 1.1.6 or later
- Servers - AS/400, Windows NT, AIX
  - AS/400 v4r2 or later, with Java and Java Toolbox installed
  - Microsoft Windows NT Server 4.0 or later with Java 1.1.6 installed
  - AIX with Java 1.1.6 installed
- Clients - IBM Network Station, Windows 95, Windows 98, Windows NT PCs
  - All require Java 1.1.6 or later

IBM

Command Line Utility Platform notes

- Configuration files on an AS/400 may be managed with the Command Line Utility running on;
  - The same AS/400
  - Any PC client on the same network
  - Any Network Station client on the same network
  - Any AS/400, Windows NT Server or AIX server on the network
- Configuration files on an AIX or Windows NT Server may be managed with the Command Line Utility running on;
  - The local AIX or Windows NT Server only
- Test configurations may be created on a PC

IBM

Command Line Utility
How does it work ?

- **Scripted configuration changes = XML reads and writes**
  - **First access to each file creates a parsed XML document in memory**
  - **Other accesses to same file read and write to document in memory**
  - **COMMIT writes all documents in memory to disk and clears memory**
    - Speeds processing
    - Other (NSM) changes after file read and before commit are over written
    - Administrator use in 'off' hours recommended
- **Scripts are written in Standard General Configuration Language (SGCL)**
  - **INSERT, UPDATE, DELETE, SELECT, COPY,COMMIT, ROLLBACK, CALL, SET, EXEC**
- **Both GUI and native os command line interfaces**

Command Line Utility Starting 1

- Java must be installed (and Java Toolbox on AS/400)
- CLASSPATH must be set
  - **May be set as environment variable (include '.')**
  - **May be set in Java startup command**
    - Can use batch file, AS/400 CL program or script
- Make .../tools/ the current directory
  - **May copy contents of .../tools/ to other directory**
- Use full package name
  - **java -classpath <classpath here> com.ibm.nsm.cl.SGCL script.txt**
    - runs script.txt from os command line (AS/400 syntax different)
  - **java -classpath <classpath here> com.ibm.nsm.cl.NSM_CL**
    - starts Command Line Utility GUI (not AS/400)

Command Line Utility Starting 2

- See readme_cl.txt for platform classpaths, settings, other
- ibmnsmcl.jar, ibmxml.jar, jt400.jar contain the application
- SGCL.ini contains user editable startup settings (optional)
- nsmcl.ini contains NSM schema and validation values
- Common problems
  - Java not installed (on AS/400, Java Toolbox not installed)
  - current directory '.' not in classpath
  - typo in classpath (application may start then lock)
  - .../tools/ not the current directory at startup
  - did not use full name - com.ibm.nsm.cl.SGCL (or NSM_CL)
  - incorrect settings, in SGCL.ini or script file
    - *PATH_TO_PROFILES, TARGET_OS, TARGET_NAME, others*

Command Line Utility Settings

- **May be set in SGCL.ini or in script file**
- **Some settings must be in order in scripts**
  - SET PATH_TO_PROFILES=/QIBM/UserData/NetworkStationV2/
  - SET TARGET_OS=AS400
  - SET TARGET_NAME=NSMTEST
- **Setting TARGET_NAME creates new server object**
  - Uses current PATH_TO_PROFILES, TARGET_OS setting
  - Used for all file access until TARGET_NAME is set again
  - Many AS/400's may be configured by resetting TARGET_NAME
  - Sign on for each not required if same User - PW exists on each
- **See SGCL.ini for examples**

Command Line Utility
Script files

- Plain text files
- One SGCL command per line
- // double slash is a full line comment
- Blank lines ignored
- Script files can CALL other script files to any depth
- Avoid recursive calls
- Create scripts to be called for standard tasks

Command Line Utility
SGCL language - select

- **SELECT**
  - **Displays and logs existing configuration values**
    - Syntax: select object/level/name/category/propertyName/
    - Example: select ibmnsm/user/joe/workstation/all/
    - The above example returns all of joe's workstation properties
    - Only properties defined in joe.nsm are returned
  - **Wild card values in name, category and propertyName fields**
    - ALL, will run one select statement for each defined value
    - ALL LIKE<regExpression>, subset of def values matching regExp
    - LIST <listFileName>, one select statement for each value in file
      - *list files have one value per line, user names or property names...*
      - *listFileName may be path and name or file name only*
      - *if listFileName is name only, then PATH_TO_SCRIPTS searched*

Command Line Utility
SGCL language - delete

- **DELETE**
  - **Removes existing configuration values**
    - Syntax: delete object/level/name/category/propertyName/ [int]
    - Example: delete ibmnsm/user/joe/workstation/all/
    - The above example removes all of joe's workstation properties
    - Only the joe.nsm file is changed
  - **Wild card values in name, category and propertyName fields**
    - ALL, ALL LIKE<regExpression>, LIST <listFileName>
  - **delete ibmnsm/user/all/all/all/ removes all user properties**
    - Files, and containing xml nodes are not removed, just properties
  - **For list properties [int] removes n'th element from list**
    - Use select to find element order

Command Line Utility
SGCL language - update

UPDATE

- **Changes existing configuration values only**
  - Syntax: update object/level/name/category/propertyName/ newValue
  - Example: update ibmnsm/system/default/desktop/help_button/ yes
  - Above example makes the help button visible on everyone's desktop
    - *unless some user or group file has a 'no' setting*
  - The allusers.nsm file is changed
- **Wild card values in name field only**
  - ALL, ALL LIKE<regExpression>, LIST <listFileName>
- **NSM 'list' type properties not allowed in UPDATE statements**
  - For list type properties use DELETE then INSERT
- **Use INSERT to create new values**

Command Line Utility
SGCL language - insert

- **INSERT**
  - **Changes or adds new configuration values**
    - Syntax: insert object/level/name/category/propertyName/ newValue
    - Example: insert ibmnsm/user/joe/desktop/help_button/ yes
    - The above example makes the help button visible on joe's desktop
    - Only the joe.nsm file is changed
      - *joe.nsm file is created if does not exist*
      - *if file is to be created, server user profile 'joe' must exist*
    - Syntax for 'list' properties: newValue has the form;
      - *-fieldName1 fieldValue1 -fieldName2 fieldValue2 ...*
      - *all fields must appear and in the correct order*
    - INSERT also sets attributes of some list properties
      - *only for 'serial-interfaces-table' and 'serial-daemons-table'*
      - *newValue may be SET ACTION INSERT (or APPEND or REPLACE)*
      - *APPEND makes list additive from various levels, INSERT additive w/ order*
      - *if INSERT attribute then following INSERT commands use STARTAT*
      - *STARTAT:2 -fieldName1 fieldValue1 -fieldName2 fieldValue2...*

Command Line Utility
SGCL language - copy

- **COPY**
  - **Copies configuration values**
    - Syntax: copy object/level/name/category/propertyName/ object/level/name
    - Example: copy ibmnsm/user/joe/ALL/ALL/ ibmnsm/user/LIST myList.txt
    - The above example copies all of joe's values to a list of users
    - All <userName>.nsm files in myList.txt are changed
      - *Each <userName>.nsm file is created if does not exist*
      - *if file is to be created, server user profile <userName> must exist*
  - **Copy from wild card values in category and propertyName field**
    - ALL, ALL LIKE<regExpression>, LIST <listFileName>
  - **Copy to wild card values in name field**
    - ALL, ALL LIKE<regExpression>, LIST <listFileName>

Command Line Utility
SGCL language - commit

- **COMMIT**
  - **Writes all pending configuration changes to disk**
    - Pending changes are all changes since last commit, rollback or startup
    - Overwrites complete *.nsm files
      - *Best to use command line during 'off' hours*
      - *Best to have only one instance of command line in use*
  - **Clears all xml parsed documents from memory**
    - Acts like a 'refresh' to *.nsm files
      - *Next file access will read *.nsm file from disk*
  - **Run scripts without commit for test and debugging**
    - Use GUI for debugging, look for errors then commit
    - Tested scripts may be run from os command line w/ commit on last line

Command Line Utility
SGCL language - rollback

- **ROLLBACK**
  - **Clears all uncommitted configuration changes**
    - All changes since last commit, rollback or program start
    - Changes that have been committed are not affected
  - **Clears all xml parsed documents from memory**
    - Acts like a 'refresh' to *.nsm files
      - *Next file access will read *.nsm file from disk*
  - **Use with GUI when errors appear**
    - Run script, see errors, rollback, edit script, run again

Command Line Utility
SGCL language - call

- **CALL**
  - **Syntax: call fileName[Path]**
  - **Example: call /myScripts/someScript.txt**
    - If any '/' or '\' characters exist then assumed to be complete path - name
    - If not then PATH_TO_SCRIPTS setting is added before name
  - **From GUI runs the named script file**
    - Same as 'Run file' button
  - **From a script file add all lines in called script at point of call**
    - Any depth of nested calls possible
    - Recursive calls create infinite loops - AVOID

Command Line Utility
SGCL language - set

- **SET**
  - **Syntax: set setting=value**
  - **Example: set TARGET_NAME=NSMTEST**
    - Same syntax from GUI or in script file
    - All settings may also be set in SGCL.ini file
      - *In SGCL.ini syntax is: TARGET_NAME=NSMTEST*
  - **Useful tool in script files**
    - Set TARGET_NAME to change servers
    - Set LOG_FILE_NAME to put different parts of the log in different files
    - Set DEFAULT_USER and rerun the same script
      - *DEFAULT_USER, DEFAULT_WORKSTATION, DEFAULT_USER_GROUP are variables that may be set and used in the name field of all commands*

Command Line Utility
SGCL language - exec

- EXEC
  - Syntax: exec <operating system command>
  - Example: exec crtusrprf usrprf(joe) usrcls(*USER)
  - AIX and Windows NT Server, only on local server
  - AS/400 both local and any number of remote servers
  - Messages returned from server are logged

Command Line Utility Errors

- **IncorrectCommandException**
  - **Wrong command syntax, value or value combination**
    - Level value does not exist
    - Property name does not exist
    - Category - Property combination not valid
    - Value out of range
- **NSMCLFileException**
  - **File not found, not readable, no file authority**
    - Stops processing of current command or script file
    - If message says no authority, user may not be admin
- **NSMCLSystemException**
  - **Unexpected program error**
    - Get full error message, script being run, platform info, refer to developer
    - Zero byte nsm files

Command Line Utility Documentation

- readme_cl.txt
  - located in the .../tools/ directory, includes help.txt (handout)
  - covers installation, setup, startup for all platforms
- help.txt
  - appears when GUI help button is pressed, also appended to readme
  - covers writing script files, language syntax
- IBM Network Station Advanced Information
  - posted on Web only at - http://www.ibm.com/nc/pubs
  - covers startup, writing script files, language syntax w/ diagrams
  - IMPORTANT: List of all property names, categories, levels, values

IBM