



CCAPRINT

A Newsletter Excerpt for System 1032 Users

April 2007

USE OF AND ACCESS TO PRODUCTS AND FEATURES ARE IN ACCORDANCE WITH THE TERMS AND CONDITIONS OF THE USER'S SOFTWARE LICENSE. THE PRESENTATION OF MATERIAL HEREIN DOES NOT, IN ANY MANNER, MODIFY SUCH TERMS AND CONDITIONS.

CHANGE Command Generator – CHGGEN System 1032 Tools & Utilities

By Tym Stegner

In my [previous article](#), I described the ADDGEN tool that I created in response to a customer request. The customer wanted the ability to create ADD commands from existing datasets. When the ADDGEN tool became a reality, they pressed me for a similar tool for the CHANGE command, since they also wanted to test per-record updates as part of their interactive test suite.

I was happy to comply, because it seemed to me that the addition of a CHGGEN tool would round out a series of related tools that included ADDGEN, DSVERIFY, and CHGGEN. Furthermore, I had already written ADDGEN--how difficult could CHGGEN be?

Designing CHGGEN

As it turned out, it was not too difficult to write CHGGEN, as it was completely different from writing ADDGEN. The ADDGEN tool converts large numbers of records into ADD commands, primarily of keyed attributes, as it is used to test fixes for key damage during updates. However, the CHGGEN tool could not just grab a selection set and convert it, because it needed user input regarding which attribute(s) to use for a FIND command, and also, CHGGEN needed input regarding which attributes to change.

The CHGGEN tool scans the list of available attributes, qualifies them for inclusion, then gathers all necessary information, such as data type, name, and format, for later command generation. This array of data drives the query and updates the user interface.

Using the DIRECTORS dataset as an example, the CHGGEN startup looks as follows:

```
1032> Open Ds DIRECTORS In S1032_DEMO Readonly  
Current Dataset Is Now DIRECTORS  
1032> Use COMPILE_CHGGEN  
S1032_HLI Is Already In Use  
1032> Call CHGGEN  
Scanning dataset...
```

System 1032 displays the following output.

Query Target?	Attribute Name	Update Target?	Data-type
Yes	\$ID	N/A	Integer
No	DIRECTOR_ID	Yes	Integer
No	FIRST_NAME	Yes	Text
No	SURNAME	Yes	Text
No	YEAR_OF_BIRTH	Yes	Date_Time
No	COUNTRY_OF_BIRTH	Yes	Text

[D]one, Edit [Q]uery, Edit [U]pdate:

The output view shows the default configuration that uses \$ID as the Query Target attribute and the other qualifying attributes as Update Target attributes. In this current release of CHGGEN, an attribute must be keyed to qualify as an update target.

Note: You can query on Special attributes, but the Special attributes cannot be updated with selection set data, as only System 1032 can (and does) automatically update them with the proper information.

Continuing the Interaction with CHGGEN

Entering **Q** at the prompt brings up the *List of attrs to FIND upon* menu:

List of attrs to FIND upon, \$ID is default	
0)	Yes - \$ID
1)	No - DIRECTOR_ID
2)	No - FIRST_NAME
3)	No - SURNAME
4)	No - YEAR_OF_BIRTH
5)	No - COUNTRY_OF_BIRTH
Enter attr# to toggle setting, 99 to exit:	

You will use this menu to select the attributes you want to base your query on.

For this example, I wanted to query using the DIRECTOR_ID, so I entered **0** to toggle the \$ID attribute to No, Then I entered **1** to toggle DIRECTOR_ID to Yes, and then **99** to return to the following View menu. The changes appear in the Query Target column.

Query Target?	Attribute Name	Update Target?	Data-type
No	\$ID	N/A	Integer
Yes	DIRECTOR_ID	No	Integer
No	FIRST_NAME	Yes	Text
No	SURNAME	Yes	Text
No	YEAR_OF_BIRTH	Yes	Date_Time
No	COUNTRY_OF_BIRTH	Yes	Text

[D]one, Edit [Q]uery, Edit [U]pdate:

I want to update only the SURNAME and YEAR_OF_BIRTH fields, so I entered **U** to access the following *List of attrs to CHANGE* menu:

```
List of attrs to CHANGE

0) N/A - $ID
1) No - DIRECTOR_ID
2) Yes - FIRST_NAME
3) Yes - SURNAME
4) Yes - YEAR_OF_BIRTH
5) Yes - COUNTRY_OF_BIRTH
Enter attr# to toggle setting, 99 to exit:
```

The DIRECTOR_ID attribute is already set to No, because DIRECTOR_ID is an Integer-of-ID data type, which users cannot update. Continuing in a similar fashion, I entered numbers to disable the attributes I do not want to update. Then I entered **99** again to refresh the View menu:

Query Target?	Attribute Name	Update Target?	Data-type
No	\$ID	N/A	Integer
Yes	DIRECTOR_ID	No	Integer
No	FIRST_NAME	No	Text
No	SURNAME	Yes	Text
No	YEAR_OF_BIRTH	Yes	Date_Time
No	COUNTRY_OF_BIRTH	No	Text

[D]one, Edit [Q]uery, Edit [U]pdate:

Now, when reading the output, I see that DIRECTOR_ID is the Query Target, and SURNAME and YEAR_OF_BIRTH are the Update Targets.

Creating a CHGGEN.DMC File

I signaled CHGGEN to continue by entering **D**.

```
[D]one, Edit [Q]uery, Edit [U]pdate: D

Generate DMC...
Created file DIRECTORS_CHGGEN.DMC
```

The DIRECTORS_CHGGEN.DMC file is the output from CHGGEN processing. This is a small System 1032 command file that takes an existing selection set and outputs the required FIND and CHANGE commands into another command file that will be used to update a copy of the DIRECTORS dataset to the values currently held in that original selection set.

Using the shorter, alternate attribute names, the DIRECTOR_CHGGEN.DMC file is as follows:

```
Init 6 DIRECTORS_CHANGE_CMDS.DMC
For Each DIRECTORS Record Do
  Write On 6 DID SN YOBB -
    Format("FIND DID " I3 " ; CHANGE SN " "' A "' ", YOBB " "' D8 "' ")
End_for
Release 6
Write Format("Created file DIRECTORS_CHANGE_CMDS.DMC...")
```

Creating a File That Stores the Changes

The WRITE command in the DMC file uses the information gathered by CHGGEN to format and output a FIND and CHANGE command per record of a precreated selection set in the DIRECTORS dataset into a file named DIRECTORS_CHANGE_CMDS.DMC.

```
1032> Find $ID Le 5
5 DIRECTORS records found
1032> Use DIRECTORS_CHGGEN
Initializing Channel 6
Created File DIRECTORS_CHANGE_CMDS.DMC...
1032> Exit
$ Type DIRECTORS_CHANGE_CMDS.DMC
FIND DID 1; CHANGE SN "Allen", YOBB '1935'
FIND DID 2; CHANGE SN "Altman", YOBB '1925'
FIND DID 3; CHANGE SN "Anderson", YOBB '1920'
FIND DID 4; CHANGE SN "Anderson", YOBB '1923'
FIND DID 5; CHANGE SN "Antonioni", YOBB '1912'
$
```

Cleaning up the CHANGE_CMDS.DMC File

Once the DIRECTORS_CHANGE_CMDS.DMC file has been generated, the CHGGEN_CLEANUP.COM command file is run against it.

```
$ @CHGGEN_CLEANUP
Enter name of DMC file: DIRECTORS_CHANGE_CMDS.DMC
%TPU-S-FILEIN, 10 lines read from file
DSK2:[CHGGEN]DIRE_CHANGE_CMDS.DMC;1
%TPU-S-FILEOUT, 10 lines written to file
DSK2:[CHGGEN]DIRS_CHANGE_CMDS.DMC;2
$
$ Type DIRECTORS_CHANGE_CMDS.DMC
FIND DID 1; CHANGE SN "Allen", YOB '1935'
FIND DID 2; CHANGE SN "Altman", YOB '1925'
FIND DID 3; CHANGE SN "Anderson", YOB '1920'
FIND DID 4; CHANGE SN "Anderson", YOB '1923'
FIND DID 5; CHANGE SN "Antonioni", YOB '1912'
```

The CHGGEN_CLEANUP command file takes care of leading spaces and leading zeros, as well as correctly adjusting MISSING values.

Getting Your Copy of CHGGEN

To order a copy of the CHGGEN tool, please apply to System 1032 Support via phone or e-mail. The CHGGEN tool is delivered via e-mail as a zipped BACKUP save-set, including instructions for unpacking and setup.

In Summary

In this article, I have described the workings of the CHGGEN tool that is used to generate FIND and CHANGE commands to test interactively updating a dataset.