



**CCAPRINT**

**A Newsletter Excerpt for System 1032 Users**

*January 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.

## **ADD Command Generator – ADDGEN System 1032 Tools & Utilities**

*By Tym Stegner*

This article describes ADDGEN, a new System 1032 utility, how it came about, how to use it, and some details of the implementation. The purpose of the ADDGEN utility is to create a DMC file of ADD commands that can add pre-existing data to a dataset.

### **A Great Idea Is Hatched**

A while back, a customer and I were discussing means of testing appending records in bulk. We wanted to test some interactive appends, such as what would happen during interactive data entry. The customer wished aloud that System 1032 had a utility to generate ADD commands from an existing dataset. Thus, the genesis of the idea for ADDGEN came about.

### **Creating the Appropriate Utility**

The ADDGEN utility is designed to create a DMC file of ADD commands from a particular dataset structure. Instead of creating a procedure to extract records to a DMC file, I opted to have the utility generate the means to DUMP a given selection set from the dataset via an RD, then post-process the dumped data into the required ADD commands. The rationale for this approach was prompted by the customer, who had revealed that the datasets under consideration for testing had record counts numbering in the tens of millions. Some experimentation with the PRINT command versus the DUMP command showed me that DUMP processing is about three times faster than PRINT processing for this sort of endeavor.

### **Understanding ADDGEN**

The ADDGEN utility takes a single parameter, that being a flag to select either all attributes in the dataset or only the keyed attributes. The so-called special attributes-- DATE, DATE\_TIME, or USER\_OF\_ENTRY, \_OF\_CHANGE, or \_OF\_DELETE are automatically omitted, as such attributes cannot be added in any case.

The procedure scans the requisite attributes and builds an RD to produce the bulk of the necessary ADD command, using the shortest of the names for the attribute to keep the record length down.

- Text items will have double quotes enclosing their values
- Date and time values will be enclosed in single quotation marks
- Numerics have no delimiters.

The RD uses the DELIMIT AFTER option from the field definition to output the delimiters, punctuation, and attribute names that occur between each value in the record. The last attribute value is followed by a trailing delimiter, if required, and a semi-colon that completes the command.

For example, consider the following SAMPLE dataset:

```
1032> Show Ds SAMPLE Definition
Dataset SAMPLE;
  Attribute XDATE Date;
  Attribute XINT Integer;
  Attribute XTEXT Text 20;
  Attribute DOE Date Of Entry;
End_Dataset;
1032>
```

An ADD command for this dataset may look like the following:

```
ADD XDATE '11/28/1960', XINT 45, XTEXT "Hello, world!";
```

If we separate out the values from the rest of the command – expecting these to be taken from the dataset, we can see the remaining text forms the basis of the AFTER delimiters:

```
ADD XDATE '11.28,1960', XINT 45, XTEST "Hello, world!";
```

That text which appears after a value can be used to form the string for the after-attribute delimiter text in the RD. The leading text of the ADD command must be handled separately.

The RD that creates this ADD command was built as follows:

```
RD SAMPLE_RD
  FIELD XDATE Date FORMAT 3 RD_MISSING "MISSING" -
    DELIMIT BY #TD|', XINT | AFTER
  FIELD XINT Integer RD_MISSING "MISSING" -
    DELIMIT BY #TD|, XTEXT " | AFTER
  FIELD XTEXT TEXT DELIMIT BY #TD|"; | AFTER
END_RD
```

Note the initial ADD command keyword, the initial attribute name (XDATE), and the initial lead delimiter (') are not found in the RD. Due to a limitation of RDs, you cannot declare a field to attach the necessary string as an AFTER delimiter. To work around this limitation, a second file is produced, known hereafter as the prefix file, which is used later in the post-processing step.

### Using ADDGEN

To use the ADDGEN utility, you open a particular dataset for which a set of ADD commands is to be generated, and compile the ADDGEN source file. ADDGEN is called with the proper parameter, and the RD and prefix file are created for that dataset, having the names <datasetname>.ADDGEN\_RD, and <datasetname>.ADDGEN\_PREFIX.

Due to the PRINT vs. DUMP efficiency previously noted, creating the ADD commands for the dataset is a two-step process:

1. A raw file of data values and after-attribute delimiters is created.
2. This raw file is post-processed into a finished DMC file of ADD commands for the dataset.

At this point, or at any point hereafter, the dataset is opened (if not already open), and a selection set of necessary records is created. If not already available, the ADDGEN-created RD is compiled. The raw file is created from the selection set using the DUMP command whose output file uses a file type of DMC and using the RD <datasetname>.ADDGEN\_RD.

Exit System 1032 to post-process the raw file using the command file ADDGEN\_APPLY\_PREFIX.COM. This command file passes the file name of the raw data, and the name of the prefix file, <datasetname>.ADDGEN\_PREFIX.

This command file uses the TPU editor to insert the prefix text from the prefix file at the beginning of each line in the raw data file, as well as carry out some clean-up of extra spaces, extraneous leading zeros, and MISSING value conversions. The raw data file from the dataset has now been converted into a DMC file of ADD commands.

### Getting Your Copy of ADDGEN

If you want a copy of the ADDGEN utility, please contact System 1032 customer support via phone or e-mail to request a copy. Delivery will be via ZIP file in an e-mail.

## In Summary

This article discussed a requested need for a utility to generate ADD commands from any existing dataset. It describes and demonstrates the process and results. ADDGEN can be used to create interactive test suites to simulate interactive update sessions to test application systems or dataset interactions. With minor editing to the ADDGEN source, it might also be used to generate INSERT commands for migration of dataset to non-System 1032 data sources.

## Coming Attractions

In our next article, the corollary to ADDGEN will be described, that being CHGGEN, a utility to create FIND/CHANGE commands from existing datasets.

*© 2007 Computer Corporation of America  
500 Old Connecticut Path, Framingham, MA 01701*