# Bulk Creation of Data Acquisition Parameters

**Benjamin Kupferschmidt**
**Technical Manager - TTCWare**
**Teletronics Technology Corporation**

## ABSTRACT

Modern data acquisition systems can be very time consuming to configure. The most time consuming aspect of configuring a data acquisition system is defining the measurements that the system will collect. Each measurement has to be uniquely identified in the system and the system needs to know what data the measurement will sample. Data acquisition systems are capable of sampling thousands of measurements in a single test flight. If all of the measurements are created by hand, it can take many hours to input all of the required measurements into the data acquisition system's setup software. This process can also be extremely tedious since many measurements are very similar.

This paper will examine several possible solutions to the problem of rapidly creating large numbers of data acquisition measurements. If the list of measurements that need to be created already exists in an electronic format then the simplest approach would be to create an importer. The two main ways to import data are XML and comma separated value files. This paper will discuss the advantages and disadvantages of both approaches.

In addition to importers, this paper will discuss a system that can be used to create large numbers of similar measurements very quickly. This system is ideally suited to MIL-STD-1553 and ARINC-429 bus data. It exploits the fact that most bus measurements are typically very similar to each other. For example, 1553 measurements typically differ only in terms of the command word and the selected data words. This system allows the user to specify ranges of data words for each command word. It can then create the measurements based on the user specified ranges.

## KEYWORDS

Data Acquisition, Bulk Parameter Creation, XML, MIL-STD-1553, ARINC-429

## INTRODUCTION

One of the key components of configuring a flight test data acquisition is the creation of data acquisition parameters. A parameter has to be created for each sensor or bus data word that needs to be sampled on a test article. Many recent flight test programs require that thousands of sensors and bus data words be sampled by the data acquisition system. This means that the engineer who is responsible for configuring the data acquisition

system will often have to create thousands of unique parameters. Due to the quantity of parameters that need to be created this can be a very time consuming process.

Instrumentation engineers often have a limited amount of time to configure a data acquisition system before a flight test. Modern flight test programs also require that the parameters that are sampled on each flight test can vary greatly. These factors mean that it is often impractical to manually create all of the parameters in the data acquisition system vendor's setup software.

There are two main types of parameters that need to be created in large quantities on most flight test instrumentation systems. The two types are bus data and orange wire sensor measurements. Most other types of data that are sampled by a data acquisition system do not require a large number of parameters. For example, FIFO based embedded data streams typically require only one parameter per stream. Audio and video data also usually requires only a single parameter to sample each data source. These parameters can be created by hand.

Most modern test articles have one or more on-board avionics buses. It is often necessary for the data acquisition system to monitor these buses. This typically requires creating distinct parameters for each data word that needs to be sampled from the bus. Some of the most common buses on test articles are MIL-STD-1553 and ARINC-429. 1553 data is organized into messages that contain data words. In some cases, the entire 1553 message which can contain up to 32 data words will need to be sampled. In other cases, only a single data word needs to be sampled from a message. ARINC-429 bus data is organized into labels. Each ARINC-429 label contains a single data word.

The other common type of data on most test articles is orange wire data. Orange wire measurements are collected from various types of sensors on a test article. These include stain, voltage, temperature, pressure and other similar physical measurements. Most orange wire measurements are collected by sensor type specific data acquisition cards. Each card collects a specific type of data and encodes it in a digital format. In some instances, these cards can collect up to 64 channels on a single card. These channels often have only a few settings and all of the channels are usually configured similarly.

Most data acquisition systems use a common word length that is typically between 8 and 16 bits per word. Each data word can only sample the number of bits in the common word length. For measurements that require more data bits, TTC's data acquisition systems provide extended read parameters. An extended read parameter outputs any additional bits of data beyond the initial common word length. Once extended reads are used, the original parameter and its extended reads need to be concatenated together to form a single measurement that is longer than the common word length. To simplify data analysis, it is typically a good idea to uniquely name each extended read parameter. When the extended reads are uniquely named, it is much easier for the data playback software to select the correct locations in the PCM format for use in the concatenations. Creating extended reads is another operation that can be tedious for instrumentation engineers.

When data acquisition parameters are being defined, it is often a good idea to assign an engineering unit (EU) conversion to each parameter. Engineering unit conversions are mathematical operations that convert the raw sensor data from a data acquisition system into useful numeric measurements. For example, a temperature card in a data acquisition system that is running at 12 bits per word will collect temperatures on a scale from 0 to 4095 counts. This raw value needs to be converted to an actual temperature by applying a polynomial EU to the raw value. The output of the polynomial is a value in degrees Celsius. The same conversion function is often used for many of the parameters in a data acquisition system. This means that the instrumentation engineer must enter the same or very similar EU conversions repeatedly when configuring the data acquisition system.

There are several possible solutions to the problem of rapidly creating parameters for data acquisition system. TTC has implemented three main solutions that provide instrumentation engineers with the ability to rapidly create parameters for their data acquisition systems. By using these methods to create many of the parameters, the instrumentation engineers will be able to configure a system much more rapidly than if the system had to be configured entirely by hand. Each solution has advantages and disadvantages and is geared towards a particular class of applications.

## USING XML TO CREATE PARAMETERS

The first method for rapidly creating data acquisition parameters uses XML to transfer parameter information from a user's internal test database to the data acquisition system's configuration software. The main advantage of using XML to transfer configuration information is its flexibility. TTC has defined a set of XML tags that allow every setting that can be configured by hand in a data acquisition system to also be configured through XML.

The flexibility of the XML solution makes it possible for instrumentation engineers to define an entire system configuration in XML or just a single parameter. If the engineer creates an XML file that describes a complete system then it is possible to simply import the file into the configuration software and generate hardware programming without having to manually configure a single setting. Alternatively, the XML could be used to create a set of 1553 or ARINC-429 parameters in an existing project where all of the other settings are defined manually.

Another advantage of XML is that it is bidirectional. The configuration of an existing project can be exported as an XML file. The user could then read the configuration into their internal database. Once the configuration is in the internal database then the user could regenerate the XML with a different set of parameters. This makes it easy to re-use parts of a configuration on a different test flight. The XML importer also provides a high level of error checking to ensure that the configuration is imported correctly and to notify the user about any errors in the configuration.

```
- <PARAMETER ShortMnemonic="AMUX-3789-DW04" LongMnemonic="AMUX-3789-DW04">
  - <SETTINGS>
      <COMMAND_HEX>0x3789</COMMAND_HEX>
      <REMOTE_TERMINAL>6</REMOTE_TERMINAL>
      <TRANSMIT_RECEIVE>Transmit</TRANSMIT_RECEIVE>
      <SUB_ADDRESS>28</SUB_ADDRESS>
      <MESSAGE_COMPONENT>Data Word</MESSAGE_COMPONENT>
      <DATA_WORD>4</DATA_WORD>
      <MESSAGE_TYPE>RT To BC Transfer</MESSAGE_TYPE>
    </SETTINGS>
  </PARAMETER>
```

**Figure 1: A MIL-STD-1553 Parameter in XML**

Despite its advantages, XML is not always the best solution for creating parameters quickly. Learning how to use XML requires a substantial time investment. The user must become familiar with the tags that have been defined for each type of parameter. It also takes longer for a user to get started with XML than it does for the other methods that this paper will discuss. In order to use XML to import configuration information, users will typically have to write a custom program that converts the configuration information from their measurement database into TTC's XML format.

XML also has limited usefulness if a user does not have an electronic database of the measurements that they want to sample. Without a measurement database, the user will need to manually create the measurements in XML. Writing XML files by hand is a difficult task because tag and attribute names must be structured correctly so that the file can be interpreted.

The only time when hand modifying an XML file can save a substantial amount of time is when a user wants to make a bulk change to a particular setting. For example, if a group of 1553 parameters are currently assigned to one sub-address and you need to move them to a different sub-address, this could be done easily with a simply find and replace operation in an XML file.

## IMPORTING PARAMETERS FROM CSV FILES

The second method for rapidly creating parameters is to import the parameters from a Comma Separated Values (CSV) or tab delimited file. This method is ideal for users who typically use spreadsheets to manage their parameter lists. A CSV file can be easily generated from any spreadsheet application like Microsoft Excel.

The main advantage of using CSV files is that they are very simple to create and maintain. Many test engineers already use spreadsheets to organize their parameter lists and decide what parameters to sample. The basic design of the CSV importer assumes that each line in the CSV file describes one parameter.

There are two different approaches to using CSV files to import parameters into the vendor's configuration software. The first approach is to define a set of columns that must be included in each CSV file. The set of columns will have some columns like

Short Mnemonic that are common to all types of parameters. The rest of the columns will depend on the type of the parameter. For example, the format for a 1553 parameter could define columns for Command Word, Remote Terminal, Transmit/Receive, Sub-Address, Word Count and Data Word Number. By defining a fixed set of columns for each type of parameter, it is simpler to perform error checking. The problem with this approach is that it may make it difficult for users who have existing spreadsheets to generate valid CSV import files. These users might need to adjust the columns in their spreadsheet or create a copy of the data and adjust it before generating a CSV file.

| | A | B | C | D | E | F | G | H |
|---|---|---|---|---|---|---|---|---|
| 1 | **Short Mnemonic** | **Command Word** | **RT** | **Tx/Rx** | **SA** | **WC** | **Sample Type** | **Data Word** |
| 2 | AMUX-3789-CW | 3789 | 6 | Tx | 28 | 9 | Command Word | |
| 3 | AMUX-3789-DW01 | 3789 | 6 | Tx | 28 | 9 | Data Word | 1 |
| 4 | AMUX-3789-DW02 | 3789 | 6 | Tx | 28 | 9 | Data Word | 2 |
| 5 | AMUX-3789-DW03 | 3789 | 6 | Tx | 28 | 9 | Data Word | 3 |
| 6 | AMUX-3789-DW04 | 3789 | 6 | Tx | 28 | 9 | Data Word | 4 |

**Figure 2: Five MIL-STD-1553 Parameters in a Spreadsheet**

The second approach to creating CSV files is to define a set of possible columns for each type of parameter. The user would then be able to create files that contained any combination of the allowable columns. To do this, the user would have to tell the import software the order of the columns. Alternatively, the importer software could use the column header row in the CSV file to automatically determine the type of each column. The main advantage of this approach is that it makes it possible for the user to make the importer work with an existing spreadsheet format.

There are several disadvantages to using CSV or tab delimited files to import parameters. The first problem is that the source spreadsheet has little or no error checking ability. This means that the user may import incorrect or invalid settings and not realize it. The importer software must check all of the imported values to make sure they are valid. If the user selects the columns to include in the spreadsheet then they may omit essential information that would have to be manually added for each parameter. It's also possible that the user may interpret a setting slightly differently than the importer. For example, the user might specify a value using a range of 1 to 20 while the importer expects 0 to 19. All of these factors can make importing a CSV file very complicated for the data acquisition configuration software.

Another potential problem is that there are certain settings for data acquisition parameters that are difficult to express in the form in a single line per parameter format. An example would be an EU with many arguments like a lookup table or a polynomial with a high order. These types of settings cannot be easily imported from a CSV file.

## USING A SOFTWARE WIZARD TO CREATE PARAMETERS

The final method for automating the creation of large quantities of data acquisition parameters is the use of a software wizard. The wizard interface asks the user a series of questions and then automatically generates a set of parameters based on the user's

answers. This approach is ideal for cases where the user needs to create large numbers of similar parameters that differ mainly in terms of the bus data word or analog channel number that they are sampling. The wizard is particularly well suited to parameters that only have a few user selectable settings.

One of the advantages of the software wizard is that the user does not need to create an import file. There is also no need for the user to write any custom software, they just follow the steps in the wizard to create the parameters. This approach is very well suited for 1553, ARINC-429 and other bus parameters. It makes it very easy to sample an entire 1553 message or all of the ARINC-429 bus labels on a particular channel. Extended read creation is also integrated into the wizard interface.

## BULK PARAMETER GENERATOR

The bulk parameter generator asks the user a series of questions and then uses the answers to generate a set of new parameters. The key question is the desired range of values for each possible parameter setting. If the user specifies a single value for each possible setting then the generator will only create one parameter. However, if the user specifies a range of values for one of the settings then the generator will create one parameter for each item in the range of values. For example, the user could specify a range of 1 to 10 for the channel number setting. This would result in the creation of 10 parameters. Each parameter would have a channel number from the range of allowable channel numbers.

The concept of specifying a range of possible values for a parameter setting can be extended so that the user can specify a range of values for more than one setting. This means that the generator will create parameters for all of the combinations of the settings. So if the two settings are channel number and a data output format with two possible choices then the user could specify channel numbers from 1 to 10 and both possible data formats. The bulk generator would create 20 parameters for this example.

Essentially the bulk parameter generator uses the concept of nested for-loops to dynamically create parameters for each possible combination of settings. Each for-loop iterates over each item in the selected range of values for one of the possible settings. This allows the system to easily create parameters for every data word in a specified 1553 command word. It could also be used to create a parameter for every 1553 data word for an entire RT address.

The first screen in the bulk parameter generator wizard asks the user for the destination card for the new parameters. The system uses the card's type to customize the remaining questions. If the user selects a card that doesn't support bulk parameter generation then they cannot continue.

The second screen asks the user how they want to name the parameters that are dynamically generated by the wizard. Parameter naming is important because the system must be able to uniquely name each dynamically created parameter. The parameter

names are generated by using a name template. The template contains a combination of fixed text and dynamic variables. The variables are things like the card's name and the value of any of the parameter's settings. For a 1553 parameter, the user can use the command word, the remote terminal, the sub-address, the word type and the data word number. The default template for 1553 parameters is "<cardname><hex>-<wt><dw>". This template generates names like "AMUX0820-DW01". This name would be assigned to a parameter on the card "AMUX" that samples data word 1 from the 1553 message with the command word 0x0820. The user can also use the symbol "<#>" to represent a unique automatically generated number. .

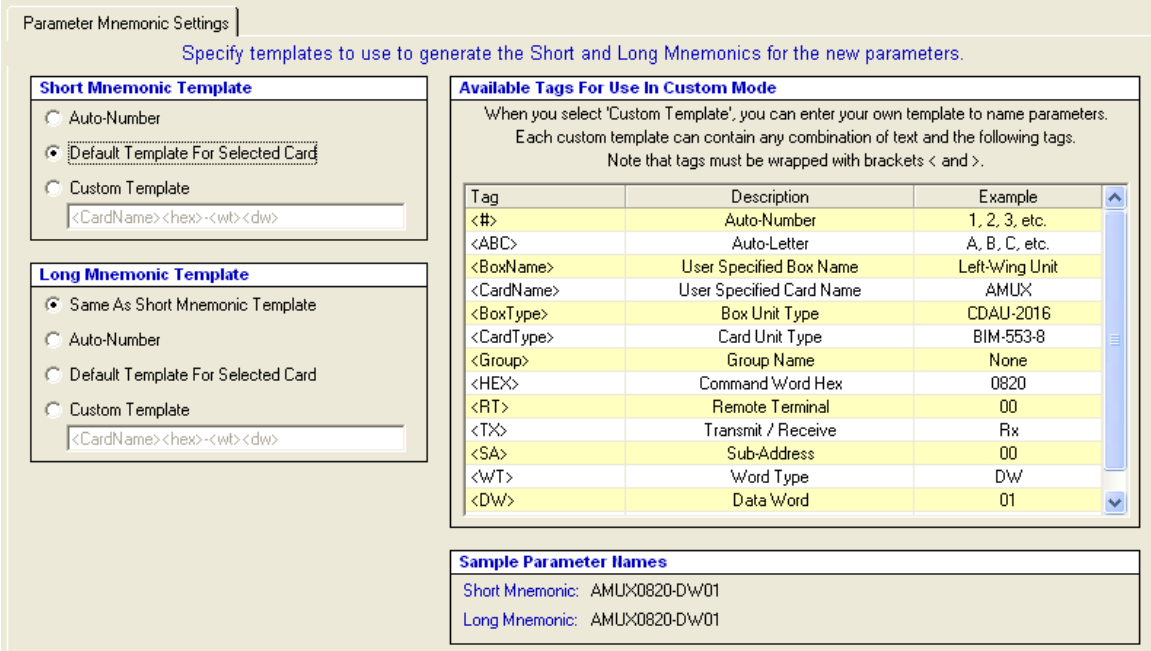


**Figure 3: Bulk Parameter Naming**

The next screen in the wizard asks the user to select the settings for the parameters that they are dynamically generating. For 1553 parameters, the allowable settings are command word, remote terminal (RT), transmit / receive, sub-address (SA), word type, data word and message type. The command word is a 16-bit hex number that contains the values for the RT, transmit / receive and SA fields. The wizard knows that these fields are linked so it automatically updates them so that they are internally consistent. For the remote terminal and sub-address fields the user can select a value from 0 to 31. For the data word field, the user can select a value from 1 to 32.

By specifying a starting and ending value for each setting, the user is able to define the set of parameters that they want to create. The system uses the nested for-loop approach to iterate through all of the possible settings and generate the parameters. Depending upon the setting, the software gives the user a variable amount of flexibility. Some settings cannot be iterated so the user is required to select a single value. Other settings allow the user to select all possible values or a single value. Other settings allow the user to pick an arbitrary range of values. A feature that adds a bit more flexibility to the wizard is the ability to specify the increment value. By selecting an increment other than

one, the user is able to create a set of parameters that sample every other data word or every third data word. This gives the user a great deal of flexibility in defining the set of parameters that will be generated dynamically.

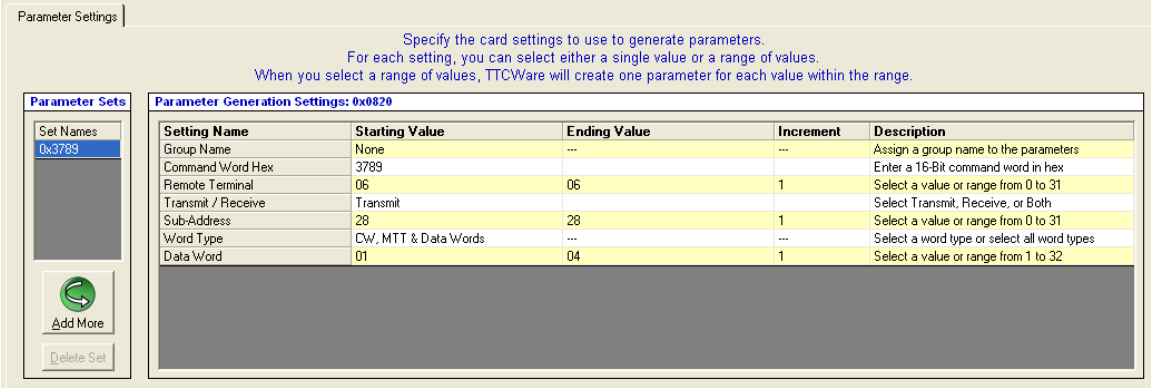


**Figure 4: Choosing a Range of Settings for a Set of 1553 Parameters**

The next screen in the wizard asks the user about extended reads. If no extended reads are required then the user can simply skip over this screen. The user can tell the wizard to create one or more extended read parameters for each real parameter. For example, if the actual data was 32-bits long and the system was running at 16 bits per word then the user would need one extended read parameter for each real parameter.

The extended read screen also asks the user how they want to name the extended read parameters. The choices are to append a letter, append a number or add some user specified text. Each extended read parameter starts with the real parameter's name and then adds the user specified text. This is done so that the extended read parameters have unique names.

The final screen in the wizard shows the user a preview of all of the parameters that will be generated. This allows to user to verify that the dynamically generated parameters were setup correctly. The user can also rename the parameters or remove one or more parameters from the set of automatically created parameters. When the user is happy with the preview then they can select the "Create All Parameters" option and the software will generate the parameters.

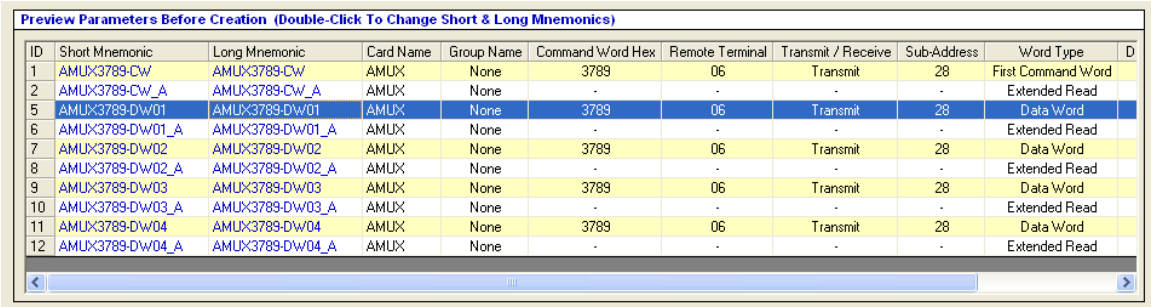


**Figure 5: Previewing the Dynamically Created Parameters**

## CONCLUSION

This paper has discussed three methods for rapidly creating data acquisition parameters. These methods help users to configure their data acquisition systems more rapidly and efficiently. XML is an ideal solution when the user already has a database containing the desired parameter settings and has the time to write a software converter to convert the database into TTC's XML format. The CSV importer is a good choice for users who define their data acquisition parameters in a spreadsheet. Finally the bulk parameter generation wizard is ideal for rapidly creating parameters that share most settings and only differ in terms of a few settings.

There are several possible enhancements that could be added to the bulk parameter generator. One would be adding the ability to assign EU functions through the CSV importer or the bulk parameter generation wizard. It would also be useful if the wizard allowed for greater customization without increasing its complexity. These enhancements would make these methods for creating parameters even more useful for users.