**4 – Using zVariables to Write Data into ABCD Files**

| **Slide #** | **Audio/Narrator** | **Graphics** |
| --- | --- | --- |
| 1 | Title page | Standard title page |
| 2 | Welcome to Module 4, “Using zVariables to Write Data into ABCD Files.”  In this module you will learn:  What a zVariable is?  What MIMOSA metadata is?  What metadata goes into zVariable attributes?  And how to use zVariables for binary data storage. | Standard introduction page |
| 3 | The power and flexibility of CDF comes from its ability to store a wide variety of binary data of differing types, sizes, and dimensions in a single record in the CDF file.  The CDF’s binary storage ability is appropriate for large amounts of sensor data because of its efficient conversion to and from is completely transparent to the user. |  |
|  | What a zVariable is? |  |
|  | What MIMOSA metadata is? |  |
|  | What metadata goes into zVariable attributes? |  |
|  | How to use zVariables for binary data storage. |  |
| 4 | The zVariable in CDF is the ***mechanism*** by which bulk data is stored into an ABCD file. |  |
| 5 | CDF can best accommodate data that can be organized into arrays.  ***A zVariable is capable of storing an array of data.***  ***Each zVariable has a data type associated with it.***  ***CDF supports all common numerical data types.***  ***A zVariable has a dimensionality and a size for each dimension.*** |  |
| 6 | ***For example, one could create a one dimension array of ten elements or a matrix of ten rows by five columns. A zVariable can hold one or more complete sets of data.***  In other words, a zVariable can have one or more records of data provided each record of data matches the original size, dimension, and type with which the zVariable was configured. |  |
| 7 | In addition to storing numeric data, a CDF zVariable supports the notion of variable level attributes, allowing each zVariable to describe its contents.  This view revisits the snapshot of our platform from which we created global attributes in (an earlier presentation) a previous module. | [1] |
| 8 | The attributes of a zVariable are used to store metadata which describes the contents of a file or variable.  Attribute exist at two levels (or scopes.)  The global scope attributes store metadata about the overall file.  ***While globally scoped attributes are discussed in an earlier module, we are not concerned with the variable scope attributes which store metadata further describing the individual variables and their values.*** |  |
| 9 | Examples of variable level attributes might include such information as:  the units in which the data is stored, a valid minimum and max threshold, etc. |  |
| 10 | In an ABCD file a zVariable contains the data for ***a single measurement location***. Its attributes will store the MIMOSA metadata which identifies the measurement location on the platform.  The interface requirement specification maintained at DISCoE describes all required metadata which must be written into the attributes of a zVariable. | [2] |
| 11 | These two items of metadata are used to identify the measurement location for the sensor or computed measure being stored. |  |
| 12 | Often a measurement location is referred to as a sensor, a measurement, a parameter, or a computation.  In addition to identifying the measurement location from which the data comes, |  |
| 13 | These attributes describe the output data type.  Data recorded from a measurement location could be:  Data Acquisition (which would be raw data), Data Manipulation, or State Information.  As mentioned earlier, zVariables are the mechanism by which data is stored in an ABCD file.  In CDF “zVariable” is a generic object that represents data.  Data can be zero dimensional (scalar) or multi-dimensional.  CDF supports up to ten dimensions in a single zVariable. |  |
| 14 | When a zVariable is created, its properties must be defined.  Properties include:  The data type stored in the zVariable.  The number of dimensions.  The size of each dimension,  And the row majority of how the data is stored.  CDF gives complete control over configuration of an array of data.  In addition to being able to specify the size of the data array the zVariable holds, a zVariable is capable of holding one or more instances of that size of data.  Each instance would be referred to as a record of data.  Before any data or attributes can be written to a zVariable in an ABCD file, the zVariable must be created. |  |
| 15 | At creation time, all of the properties mentioned earlier must be specified.  Optionally, the user can specify a compression type to be applied to this zVariable.  In addition to configuring the zVariable, since the zVariable will have attributes, those attributes must be created before they can be written for each zVariable. |  |
| 16 | If zVariables share an attribute with the same name, that particular attribute needs only to be created once. |  |
| 17 | That would be the case of all the metadata required for each zVariable.  This is a conceptual view of zVariables in an ABCD file. |  |
| 18 | In this situation, each zVariable is configured to hold a scalar item of data.  This is showing that there are several records written into each zVariable.  Another CDF file may hold a collection of zVariables that each holds a matrix of data.  Again, this view is showing that there could conceptually be several records written into each zVariable. | [3] |
| 19 | In this view the zVariables have different sizes.  Some are matrices, some are arrays, and some could be scalar data.  This shows how an ABCD file can collect zVariable of different sizes & types.  This view is similar to the view of a sample ABCD file used in the earlier presentation about global attributes.  In the global attribute section of the ABCD file  global attribute are shown here. | [4]  [5] |
| 20 | Some of the global attribute are left out in the interest of space.  In addition to the global attributes in the file, each zVariable is shown with its variable level metadata specified. This collection of metadata required by ABCD identifies the source that the data stored in the zVariable comes from. CDF supports a wide variety of data types. All data types are based on an eight bit byte. The size of an element of a data type is the same regardless of the computer or operating system being used. Below are a few of the common integer data types used in CDF. Several floating point sizes are supported in CDF. CDF also supports the writing of character data, and the user may specify that it is type character and also the number of characters stored in the value. There are two time-stamp data types provided by CDF. ABCD does not require the use of the “EPOCH” data type to store time-stamp information. ABCD’s standard uses a character string to store a time-stamp in GMT format. “EPOCH” values are just presented here for your convenience. | [6]  [7]  [8]  [9] |
| 21 | This completes the module “Using zVariables to Write Data into ABCD Files.  In this module you should learned:  What a zVariable is?  What MIMOSA metadata is?  What metadata goes into zVariable attributes?  And how to use zVariables for binary data storage.  You may either review parts of the information, or move on to the next module.  `  Click next to continue. | Standard closing page with contacts and comments information. |