Mass Properties Exchange Format
Definition and Usage

by

Gerald W. Fleck

i.e. Solutions, inc.
3742 Saddle Drive
Carlsbad, CA 92010

☎ (760) 434-0115
Fax (760) 434-0116
mpex@i-e-Solutions.com

Mass Properties Exchange Format
Definition and Usage

Introduction

Computer tools continue to improve, enhancing our ability to wrangle large quantities of information into forms of organized intelligence, quickly and easily. However, this creative endeavor of organizing information has commonly left the community with many different forms or expressions of the information. Most companies have databases they would like to be able to share with others they are teamed with or report to, but their systems are incompatible. We can understand the various expressions; we just can't unilaterally share in the fruits of others' labor. That was until now.

Now we have the vehicle that holds great promise to free us from the enslavement of different database formats. This vehicle is the MPEX format. You might say, this is just another format to contend with, introducing yet more diversity to a world which seeks to establish an accord. A bone to be picked, yes, but once cleaned will become the tool of the gods. Well, maybe not gods, but at least darn good mass properties engineers.

Overview

The MPEX format, which stands for Mass Properties Exchange, has resulted from working with many different organizations and coming to terms with the diversity of the formats currently used in tracking mass properties and related information. It is unrealistic to expect that the various companies will adopt a particular format when the cost to convert over to it far outweighs the immediate, near term benefit. This is where a key feature of the MPEX format comes into play. It is not necessary for companies to change their current tools to use the MPEX format. The intent is that the MPEX format can be easily created from existing data formats using readily available tools.

The tools are spreadsheets and text editors. With these, one can take existing ASCII text data and edit it to conform to the prescribed MPEX standard presentation of the data. In the process of opening a text file in today's spreadsheets, one can use an automated feature that will interpret the text data and setup the fields of information into columns within the spreadsheet. The original text data can be in a fixed space or delimited format. Once the data is brought into the spreadsheet, it can easily be moved around so that the fields appear in a specific order. Then additional data set designators can be added to identify the MPEX data formats. Data sets allow the support of different formats of data within a single MPEX file.

The MPEX format also addresses the issues of different systems using different fields, some tracked by one company, but not another. Even when companies track the same information, they commonly will use different field widths, which also can present basic incompatibilities between formats. This is handled via the MPEX format by saving the resulting file created in a spreadsheet or text editor as a tab delimited text file. Then when a field carried by the MPEX format isn't supported by a particular company's data, a blank column is simply inserted in the spreadsheet to serve as a place holder. The MPEX translator will handle the rest. Since the fields are tab delimited, it doesn't matter how long the fields are. The application reading the MPEX formatted data will only use as many characters as it supports, truncating fields as required.

Another common need is the ability to express hierarchical relationships as found in drawing trees in the data format. Some systems use leading spaces or periods to indicate the indentured nature of the data. Others use numeric coding like those encountered in the AN Codes used for functional coding. Both of these approaches are supported by the MPEX format.

The MPEX format also must be flexible, having an ability to adapt to new requirements while maintaining compatibility with accepted and established standards. It also must support different types of data, from material lists, with or without densities, to drawing tree breakdowns with local coordinate system definitions.

The principal idea behind the MPEX format is to establish a format that can be called a standard, one that offers ease of use, specific yet flexible structure and that we can all agree is an intermediate format. By intermediate I mean one that goes between your application and another application. Applications can continue to store the data the way they do and for those who want to exchange mass properties related information, they can create a MPEX format reader for translating the data into their application. The actual creation of the MPEX formatted files is performed by reading an application's published reports or raw data (if in text format) into a spreadsheet or word processor and then saving it as a tab delimited file. For those who often need to create MPEX formatted data, they may choose to write the routines that export data directly from their application into the MPEX format. Those who have an interest in accessing the data will create a MPEX translator to read the desired information and translate it into their format for their needs.

Specifics

So now that you know the basics, i.e. that the MPEX format is simply a tab delimited text file that has some specific data sets, what is a data set? A data set is made up of two sections of information. The first section is a data set designator and the second section is the actual data in a prescribed format.

The data set designator is a single line of information that is comprised of two components: a data set delimiter and a data set label. The delimiter is a specific stream of characters that tells the interpreter or MPEX translator that the format of the following data is about to be specified. The data set label is a stream of characters that identifies the type of data, i.e., what the fields are and the order that they appear. The label is usually a one or two word description.

This document defines the data set labels that currently make up the MPEX format. New data sets can be created for other types of data not covered. If the data is of a generic nature we may want to add a new data set label to this document to cover it. Making a specific label part of the MPEX format will reserve the data set label for the specified purpose. Otherwise, one can create unique data sets for specific purposes, being careful not to use any of the existing data set labels. When an MPEX translator doesn't recognize a data set label, the data will be skipped until a recognized data set is provided.

<u>Data Set Structure:</u>
delimiter label parameters
record data

Some data sets define parameters for the data set on the same line as the data set designator. When a data set has parameters associated with it the label will be followed by the tab-delimited parameters. In most cases the data of the data set begins on the next line after a data set designator. However, for some data sets, there are only parameters given.

The data section of the data set repeats record after record until either a new data set designator is given or the end of the file is reached. In the MPEX format, a data set ends with a line feed character. Unless a new data set designator is given at the beginning of the next line, the next line is assumed to be of the same format as the previous.

An MPEX formatted file can consist of any number of the available data set types. In most cases, a data set type will only occur once in a given file. In some cases, however, a data set may occur multiple times within a file separated by other data set types. The various types of data sets are presented and discussed in the MPEX Data Set Definitions section of this document.

The remainder of this section presents an example data set designator for the most commonly used data set, provides information on embedding comments in the data file, discusses how hierarchical relationships are supported in the MPEX format, and explains what the order of the data sets should be and what to do when data sets or data set fields have no accompanying data. There also is a discussion given on extended category descriptions, which at this point probably means nothing to you but will be of interest if you will be converting data from a flat file system to a fully relational database.

Data Set Designators

All data sets begin with a data set designator and they all share a common delimiter. This delimiter is made up of two words separated by a tab character. These two words are MPEX and DATASET. They need to appear as shown with all capital letters as the first two words on a new line. The data set label, which varies for each data set type, follows the second word, DATASET, and is separated by a tab character. An example of the most commonly used data set is as follows:

```
MPEX          DATASET     ENTRIES
```

This data set designator would then be followed by the data that matches the ENTRIES format. The ENTRIES data set is comprised of the drawing number (ID), description, configuration, effectivity, quantity, weight, center of gravity, moments and products of inertia, weight status, material, overtyped status and color. Additional fields can be user defined for specifying things like uncertainty and functional coding, for example, using the ADD FIELDS data set. The ADD FIELDS dataset will be discussed later. The details of the ENTRIES data set, along with all other data sets, are defined in the MPEX Data Set Definitions section.

Embedding Comments

Comments can be embedded in the MPEX file in several different ways. Beginning the first field with a semicolon (;) or by leaving the first field blank will cause the line to be treated as a comment line. Also, any lines that precede the first data set designator, or those that follow a data set designator that uses an unknown data set label will be treated as comment lines.

Hierarchical Relationships

The MPEX format supports hierarchical relationship definitions for a number of fields. These fields are the drawing number (or ID) and what the MPEX format refers to as Class Fields. The field for which hierarchical relationships are being defined always appears as the first field in the data set format. So, as with the ENTRIES data set, the drawing number is the first field, and in this position, it can be used to indicate parent-child relationships.

There are two methods supported by the MPEX format for indicating parent-child relationships. The first uses indentation levels. In this method, the child indented relative to its parent by a space, a colon, or a period indicates the parent-child relationship. There is no limit to the number of levels that can be indicated with this method. The second method uses a numeric code like those used for AN coding (Army Navy function codes). The numeric method supports using up to an eighteen-digit code for drawing numbers (ID) and class fields. Each set of two numbers in the code corresponds to a level.

Therefore, the data can have up to nine levels for drawing numbers (IDs) and class fields . Let's look at some examples of these very different hierarchical relationship definition methods.

Indentation Method	**Numeric Method**
Drawing number (ID) field	Drawing number (ID) field
Super Dupper Project	0100000000
.SD Installation One	0101000000
..SD Assembly One	0101010000
...SD Part One	0101010100
...SD Part Two	0101010200
..SD Assembly Two	0101020000
...SD Part Three	0101020100
...SD Part Four	0101020200
.SD Installation Two	0102000000
..SD Assembly Three	0102010000

Indentation Method	**Numeric Method**
Class One (AN Code)	Class One (AN Code)
020000 Aerodynamic Surfaces	020000 Aerodynamic Surfaces
.020100 Aero Surface-Wing	020100 Aero Surface-Wing
..020106 Covering	020106 Covering
..020108 Spars	020108 Spars
..020109 Ribs	020109 Ribs
.020300 Horizontal Stabilizer	020300 Horizontal Stabilizer
..020331 Covering	020331 Covering
..020333 Spars & Stiffeners	020333 Spars & Stiffeners
..020334 Ribs & Doublers	020334 Ribs & Doublers
..020345 Fairings & Fillets	020345 Fairings & Fillets
..020347 Exterior Finish	020347 Exterior Finish
..020349 Fittings	020349 Fittings
.020500 Vertical Stabilizer	020500 Vertical Stabilizer
..020531 Covering	020531 Covering
..020532 Spars	020532 Spars
..020534 Ribs	020534 Ribs
030000 Aerodynamic Surfaces	030000 Aerodynamic Surfaces
.030100 Surface-Elevons	030100 Surface-Elevons
..030106 Covering	030106 Covering
..030107 Spars	030107 Spars
..030109 Ribs	030109 Ribs
..030113 Joints Splices & Fast	030113 Joints Splices & Fast

The indentation method examples shown use a period. Leading colons or spaces can be used if present in the data instead of the periods. Colons and periods are usually a lot easier to visually inspect and discern.

You'll also notice that extended descriptions are given for AN codes. This is something you'll want to use if you're moving the data from a flat file system to a fully relational database. The idea is that you can still designate the data by its six-character code in the data field but define an extended description for it in the data set that defines the relationships for the class categories. This is explained in the subsequent Extended Category Descriptions section.

The data sets that support hierarchical relationships default to using the indentation method. When the numeric method is used the label has an asterisk appended to it, e.g. ENTRIES*. This informs the MPEX interpreter that numeric coding method is used.

Grouped Relationship Data

The MPEX format supports a special type of hierarchical relationship definition used for creating grouped information within some data sets. This grouping capability works similarly to the indentation method described above but is limited to a single level. For group definitions, the group member relationships are always indicated by the member being indented relative to its group by a space, a colon, or a period. The other unique aspect of group definitions is that the data set fields of the group are usually different from those used for the member in the relationship.

Unused Data Sets

The MPEX file should only include data set designators for the data sets with the associated data. Do NOT include a data set designator unless it is followed by the corresponding data.

Unused Fields

In many instances you will not have data for all the fields in a particular data set. When this occurs you can simply leave the fields blank. The important thing is that blank fields are given if fields to their right contain information. For example, in the ENTRIES data set the first nine fields in the order that they appear are the drawing number (ID), description, configuration, effectivity, quantity, weight and center of gravity (x, y, and z components). If the only data you have is drawing number (ID), description, quantity, weight and center of gravity, then you need to insert blank fields for the configuration and effectivity. The inertia fields and others that follow after the center of gravity in this data set can be left blank. Blank columns that would otherwise appear at the end of data set records need not be specifically defined. The MPEX interpreter must be smart enough to recognize this situation so that you don't need to worry whether the spreadsheet will insert trailing tab characters when the file is saved as tab delimited.

Adding Fields to Pre-Defined Data Sets

A special data set is used to extend a data set definition to include additional fields. The data set label, ADD FIELDS, identifies the data set used for this purpose. Additional fields can be added to any pre-defined data set through the specification given with the use of this data set. All fields added will appear in the order that they are defined after the fixed fields provided by the data set being modified.

Extended Category Descriptions

It is common for older mass properties applications to track things like material or source by using cryptic one or two letter codes. Also the page|column|row designations are commonly used for indicating a particular category of the AN code (Army Navy functional coding). These methods have become outdated by fully relational databases that can efficiently support long descriptions. A feature of the MPEX format is the support that it provides in automatically converting cryptic abbreviations to fully described categories.

The CLASS data sets have been set up with this support. The way it works is that the CLASS data set data can use the extended descriptions to define the categories (and their relationships to each other) and then the class fields or columns in the ENTRIES data set can use abbreviated versions of the full description for the categories. In the CLASS data set use a space to separate the extended description from the numeric portion.

The logic of the MPEX translator is to first check if an exact match of the ENTRIES class field data is already defined. If an exact match exists, the translator uses or references the one that is already defined. If an exact match doesn't exist, the translator checks to see if a category exist that starts with the same description as the abbreviated one. If that exist, the translator uses or references the category that starts with the common string of characters. Otherwise, when no match is found, the category given by the ENTRIES class field data is added or inserted as a new category.

An example of this usage would be when the ENTRIES data set has the AN Code as one of the added class fields. The data could be something like 030113 but, if in the CLASS data set for that field the same category was given as 030113 Joints Splices & Fasteners, the MPEX translator for a relational database would reference this longer description of the category for the particular entry's data.

Ordering of the Data Sets

The MPEX format is sensitive to the order in which data sets are given in the MPEX file. Typically the data set that defines the data categories of a field should appear before the particular field data is referenced by another data set. For example, if the ENTRIES data set includes data on the weight status, the WEIGHT STATUS data set should appear earlier in the file than the ENTRIES data set. This way the order in which the weight status categories are inserted into the database will be controlled by the WEIGHT STATUS data rather than in the order that the categories are referenced or used in the ENTRIES data. This becomes even more important when working with the class fields. You always want to define the hierarchical relationships of categories, using the CLASS data set, before they are referenced by another data set. If a category and its relationships for a CLASS field are not defined prior to its use, it is inserted as a top-level category.

Furthermore, the UNITS data set should be specified prior to the use of any data sets that have fields that would be dependent on the units specified.

MPEX Data Set Definitions

This section presents the data set definitions for all of the currently defined data sets. They are given in the order that they might appear in an MPEX file. The data set designator is given for each, proceeded by a comment line as it might appear in the MPEX file. The comment line gives the field descriptions for the following data set.

Description

```
; MPEX    DATASET    DESCRIPTION    Description    Path               Catalog File
MPEX      DATASET    DESCRIPTION    description    D:\dbmass\data\    DESCRIPT.CAT
```

The DESCRIPTION data set describes the source of the data, providing its description and the path and catalog file for the source database. This data set is generally not required but is recommended for informational purposes and should appear at the beginning of the file prior to any data.

The DESCRIPTION data set format follows the same format as other MPEX data sets. The data set label follows the data set delimiter. This data set has parameters but has no data that follows on a subsequent line. The description data is handled as parameters to the data set and, therefore, follows the data set designator on the same line. A TAB character separates each parameter.

Units

```
; [mass]:        OZ      LB      SLUG  G       KG      M.T.    S.T.    L.T.
; [length]:      IN      FT      MM      CM      M
; [delta time]:  SEC     MIN     HR      DAY
MPEX      DATASET      UNITS      [mass] [length] [delta time]
```

The UNITS data set is used to specify the units of mass, length and/or delta time that are used in the data file. This data set should occur prior to any data that may use the units specified. Normally the units would be specified once for a data file. However, this data set can occur multiple times as needed to deal with mixed units data files.

The UNITS data set format follows the same format as other MPEX data sets. The data set label follows the data set delimiter. This data set has parameters but has no data that follows on a subsequent line. The units data is handled as parameters to the data set and, therefore, follows the data set designator on the same line. A TAB character separates each parameter.

The brackets [] are used to show that the data is optional (do not include brackets in the data).

Mass can be any of the following abbreviations:

OZ	LB	SLUG	G	KG	M.T.	S.T.	L.T.
ounces	pounds	slugs	grams	kilograms	metric tons	short tons	long tons

Length can be any of the following abbreviations:

IN	FT	MM	CM	M
inches	feet	millimeters	centimeters	meters

Delta time can be any of the following abbreviations:

SEC	MIN	HR	DAY
seconds	minutes	hours	days

Change Case

```
; MPEX           DATASET       CHANGE CASE        alpha case      numeric case
; options:       NONE    LOWER      UPPER       TITLE       PASSIVE
MPEX             DATASET       CHANGE CASE        NONE            NONE
```

The CHANGE CASE data set is used to specify whether the MPEX translator should change the case of the text given in the data file. This data set is not required but can prove useful if you wish the text data case to be changed. This data set should occur prior to any other data set that it is intended to affect. This data set can occur multiple times, as needed to deal with different data sets.

The CHANGE CASE data set format follows the same format as other MPEX data sets. The data set label follows the data set delimiter. This data set has parameters but has no data that follows on a subsequent line. A TAB character separates each parameter.

Two parameters are used to specify how text is to be treated when it is comprised of pure alpha or alphanumeric characters. The first parameter will apply to text that contains no numbers. The second applies to text that contains some numbers and alpha characters. The options for both parameters work as follows:

Option	Description
NONE	No change of case requested
LOWER	Change all characters to lower case
UPPER	Change all characters to upper case
TITLE	Change the first character of each word to upper case and the rest to lower case
PASSIVE	Leave the first character of each word as is and change the rest to lower case Note: Passive is short for Passive Title.

Weight Status

```
; Description    Factor (%)
MPEX             DATASET      WEIGHT STATUS     [PURGE]
```

The weight status is an ENTRIES field that can be used to compute a weight margin based on the defined factor. It is the field used for weight maturity reporting. This data set is used to define the weight status categories and their associated factor, e.g. Calculated 0.15.

This data set has one optional parameter which is separated from the label with a TAB character. The parameter, PURGE, when provided is used to indicate that any previously defined weight status categories should be deleted prior to adding the new ones provided.

This data set supports defining groups. To define a group, list the group description and each member of the group using the indentation method described for creating hierarchical relationships. Groups are limited to one level. In other words, a group cannot be a member of another group. The groups in this data set would not have a factor value assigned.

Uncertainties

Description, link status, (+/-)weight, (+/-) X, (+/-) Y, (+/-) Z, (+/-) Ixx, (+/-) Iyy, (+/-) Izz, (+/-) Ixy, (+/-) Ixz, (+/-) Iyz, (+/-) mode

```
MPEX             DATASET      UNCERTAINTIES
```

Due to the number of fields associated with this data set, which causes them to extend beyond the width of this page if placed on a single line as they might appear in a comment line, they are simply listed above in the order that they would appear.

The UNCERTAINTIES data set provides a list of uncertainty factors uniquely identified by their description. The uncertainty values can be assigned to the lowest level entries (parts) by the associated description. The link status indicates whether the uncertainty values are linked to the corresponding weight status with the same description. This field is either 1 (true) or 0 (false). The weight has units of mass; the x, y, z values have units of length; and the remaining values have units of inertia, except for the mode field. The mode indicates whether percentages are used for the weight and inertias or actual values are given. Percentages are indicated with a value of 1 (true) and actual values are indicated with a value of 0 (false).

This data set has no parameters.

Materials

```
; Description   Density
MPEX            DATASET       MATERIALS        [PURGE]
```

The material is an ENTRIES field that can be used to calculate the volume of particular part and can be used for tracking material usage. This data set is used to define the material types and their associated densities. The units of density are those specified by the units data set in mass/length$^{3.}$

This data set has one optional parameter which is separated from the label with a TAB character. The parameter, PURGE, when provided is used to indicate that any previously defined materials should be deleted prior to adding the new ones provided.

This data set supports defining groups. To define a group, list the group description and each member of the group using the indentation method described for creating hierarchical relationships. Groups are limited to one level. In other words, a group cannot be a member of another group. The groups in this data set would not have a density value assigned.

Add Fields

```
; MPEX          DATASET       ADD FIELDS  dataset label
; Description   Type          Width       Units of
; [MPEX UNCERTAINTY]
MPEX            DATASET       ADD FIELDS  dataset label
```

The ADD FIELDS data set is used to add fields to those that are predefined by any particular data set. The fields that are defined by this data set are added to the right of the predefined fields of the data set being modified in the order given in this data set. The label of the data set being modified should appear as the first parameter after the ADD FIELDS data set label. This data set needs to come before the data set that it modifies.

The ADD FIELDS data set format follows the same format as other MPEX data sets. The data set label follows the data set delimiter. This data set has one parameter (the data set label of the data set being modified) and subsequent lines provide the added field definitions. A TAB character separates the data set labels.

Each added field must provide a description for the field, the type of field that it is, its width and, if applicable, the units the field data is given in. The following types of fields are supported:

Type	Description
ALPHA	Text field
CLASS	Class field that support hierarchical categories
NUMERIC	Numeric value field
SUMMATION	Numeric value field assigned at lowest level and summed up at higher levels
FACTOR	Descriptive field which has an associated factor at lowest level and acts as weighed average field at higher levels
CALCULATED	Calculated value based on other fields
MPEX UNCERTAINTY	Special type that defines the mass properties uncertainty fields which includes weight, x cg, y cg, z cg, Ixx, Iyy, Izz, Ixy, Ixz. Iyz, and mode when added to the ENTRIES data set and x cg, y cg and z cg when added to the ASSEMBLIES data set. All values are +/-.

The width field is the maximum number of characters allotted for ALPHA, CLASS and FACTOR fields and is either FLOAT or DOUBLE for NUMERIC, SUMMATION or CALCULATED field types.

The “units of” field can be any of the following unit types: None, Mass, Length, Inertia, Density, Percentage, Angle, Delta Time, Money or Labor Rate.

Class Categories

```
; Description
MPEX        DATASET     CLASS       Description
```

The MPEX format supports class fields through the use of the ADD FIELDS data set. Each class field is uniquely identified by its description. Each class field has independently defined categories. The CLASS data sets support defining the hierarchical relationships of the class categories. The data set has one parameter that provides the description of the class field being specified. Append an asterisk to the data set label when using numeric coding to establish parent-child relationships.

The CLASS data set can be used to give extended descriptions for the categories in the case where the original data given in the referencing data set (e.g. ENTRIES) uses cryptic abbreviations. This feature was implemented to support converting flat file data that references numeric codes or abbreviations to identify specific categories of information to relational databases. In a relational database, the category is defined in one place and references are made to it in the instances where it is used, minimizing the duplication of data. Thereby, extended descriptions for the categories can be used with minimal impact on overall database storage requirements.

The CLASS data set should appear after the ADD FIELDS data set and before the ASSEMBLIES and/or ENTRIES data sets.

Factors

```
; Description   factor
MPEX        DATASET     FACTORS     Description
```

The MPEX format supports factor fields through the use of the ADD FIELDS data set. Each factor field is uniquely identified by its description. Each factor field has independently defined factors. The data set has one parameter that provides the description of the factor field being specified.

Factor fields can be associated with ENTRIES data set and are used in calculations. This data set is used to define the factor types and their associated value. The units would be those specified by the units data set as defined for the field.

This data set supports defining groups. To define a group, list the group description and each member of the group using the indentation method described for creating hierarchical relationships. Groups are limited to one level. In other words, a group can not be a member of another group.

Standard Parts

```
;Id (part number)            Description     Weight
MPEX                         DATASET         STANDARD PARTS
```

The STANDARD PARTS data set is used to specify entries that are standard parts. This data set does not use any parameters. Use this data set before the ASSEMBLIES and/or ENTRIES data sets. The entries listed in the STANDARD PARTS data set will be interpreted as standard parts when listed in these two data sets. The units used for weight would be those defined for units of mass, as provided by the UNITS data set. This data set also supports defining "filters" which can be used to identify standard parts.

The import translator uses filter criteria to qualify entries as standard parts. A filter is specified by giving an Id and/or a Description which ends with the asterisk character (*) and a maximum weight. The filter criterion specifies what characters the Id/Description must start with. The characters preceding the asterisk are used. If only the asterisk character is given for either the Id or Description, all Ids or Descriptions will pass the filter criteria for the corresponding field. If no asterisk is given in either the Id or Description field, an exact match of the Id/Description must be given to pass that field's filter criteria. All fields must pass the specified filter conditions to be accepted as a standard part. The following example illustrates the use of filters:

```
;Id         Description     Weight
MPEX        DATASET         STANDARD PARTS
sp*         *               0.1
```

If an entry's Id starts with "sp" and its weight is less than or equal to 0.1, the entry will be interpreted as a standard part.

Adding additional rows of specifications can specify multiple filters. Multiple filters are "ORed" together. For the following example:

```
;Id         Description     Weight
MPEX        DATASET         STANDARD PARTS
sp*         *               0.1
*           Bolt*           0.1
```

Any entries that have an Id that starts with "sp" or a Description that starts with "bolt" and weighs less than 0.1 will be interpreted as a standard part. The filters are not case sensitive.

When a particular entry meets all criteria of any of the specified filters, it will be treated as a standard part. However, if the entry is an assembly with sub-entries, it will be converted to an assembly.

Assemblies Coordinate Systems

Id (drawing no.), Description, Configuration, Effectivity, Quantity, X-Origin, Y-Origin, Z-Origin, X-Pt on X, Y-Pt on X, Z-Pt on X, X-Pt on Y, Y-Pt on Y, Z-Pt on Y, About Z, About Y, About X, Scale X, Scale Y, Scale Z, CG locate, Disconnect.

```
MPEX          DATASET      ASSEMBLIES         [Entry Type]
```

Due to the number of fields associated with this data set, which causes them to extend beyond the width of this page if placed on a single line as they might appear in a comment line, they are simply listed above in the order that they would appear.

The ASSEMBLIES data set is primarily used for defining drawing tree type assembly breakdowns when entry mass properties are given in different coordinate systems. Each record in this data set identifies a particular entry's relationship to its parent. This data set supports hierarchical relationships being defined for the entries with indentation as the default relationship method. Append an asterisk to the data set label when using numeric coding to establish parent-child relationships.

The ASSEMBLIES data set has one optional parameter. This parameter can be used to specify the top-level entry type. The options are PROJECT, INSTALLATION, ASSEMBLY or SEQUENCE. A TAB character separates the entry type from the data set label.

This data set supports defining the coordinate transformation between a child and its parent by one of two methods. Both methods give the location of the child coordinate system relative to its parent and assume a right hand coordinate system definition is used. The first is the three-point method. By specifying the location of the Origin, a Point on X and a Point on Y, the relationship is fully defined. Each of these points specifies the location of the child coordinate system points in reference to the parent coordinate system. The second method uses the origin and three rotation angles. The Origin is simply the offset of the child coordinate system, and the rotation angles specify how to move from the parent to the child coordinate system. About Z gives the rotation about the parent (reference) z-axis. About Y is the rotation about the y' axis, which results from the z rotation. About X is the rotation about the x" axis, which results from the y rotation. The scale factors can be used to represent a left-hand coordinate system. If used with points, set scale to 1, 1, -1 to indicate z is left-handed z.

Once relationships are defined in this data set, there is no need to redefine them in the following ENTRIES data set. In the ENTRIES data set, the mass properties for the entry should be given in the local coordinate system, defined by the ASSEMBLIES data set.

There are two fields given after the coordinate transformation information. The CG Locate and Disconnect fields are logical data fields with a value of either a 0 (false) or 1 (true). A true value for the CG Locate field means the entry position is set to float so that its CG is located at the parent’s CG. A true value for the Disconnect field means the entry is temporarily disconnect from the parent so that its mass properties are not included.

The ASSEMBLIES data set supports using the MPEX UNCERTAINTY fields when specified with the ADD FIELDS data set. Only the X, Y and Z location uncertainty fields apply since this is a location specific dataset.

Entries

Id (drawing no.), Description, Configuration, Effectivity, Quantity, Weight, Xcg, Ycg, Zcg, Ixx, Iyy, Izz, Ixy, Ixz, Iyz, Weight Status, Material, Overtyped, Color

MPEX DATASET ENTRIES [WITH]

Due to the number of fields associated with this data set, which causes them to extend beyond the width of this page if placed on a single line as they might appear in a comment line, they are simply listed above in the order that they would appear.

The ENTRIES is the primary data set that defines the bulk of the data. In many cases, it may be the only data set in a MPEX file. This data set brings together all the data typically defined for a mass properties database. It supports hierarchical relationships being defined for the entries with indentation as the default relationship method. Append an asterisk to the data set label when using numeric coding to establish parent-child relationships.

The ENTRIES data set has one optional parameter. This parameter is used to specify whether the mass properties include the margins calculated using the weight status assignments. The default is to assume that the mass properties do not include any calculated margin. If the mass properties do include the margin calculated using the weight status percentages, then the parameter should be set to WITH to indicate “with margin” included. This flag affects all entry mass properties.

The mass properties in the ENTRIES data set will be in global coordinates unless the ASSEMBLIES data set was previously specified with different coordinate systems. The inertias are always given about the CG of the particular entry. In cases where the ASSEMBLIES data set is used to establish a local coordinate system for the entries, the ENTRIES data set should just include the entries that have no children, i.e., lowest in the hierarchy.

This data set also supports weight status, material, overtyped status and color fields. The weight status and material fields have been discussed previously as part of the corresponding data sets. In this data set these fields will contain the weight status or material information associated with a specific entry. The descriptions that are listed in these fields will be added to the database, if not previously defined, in the order in which they appear. The overtyped status field is a logical field that is used to indicate whether the mass properties of the entry are calculated values or were hand entered (overtyped). Overtyped status of 0 (false) means the values are calculated, and a value of 1 (true) is used when the entry mass properties are overtyped or forced to specific values. The color field is a numeric field used to specify color.

The ENTRIES data set supports using the MPEX UNCERTAINTY fields when specified with the ADD FIELDS data set. All of the mass properties uncertainty fields given in the ADD FIELDS data set are added to ENTRIES data set when specified. All values are +/-. The mode value is used to indicate four different logical status. These logical states are as follows:

+/- Mode Value	Description
0	actual values (manually assigned if a part, or calculated if an assembly)
1	percent values for weight and inertias manually assigned

2	automatically assigned actual values using linked uncertainties
3	automatically assigned percent values using linked uncertainties

Various fields can be added to this data set through the use of the ADD FIELDS data set. When class fields are specified, the categories are automatically added to the class categories at the top level if not previously defined in the CLASS data set for the corresponding field. As mentioned in the earlier discussion on Extended Category Descriptions, the data (categories) specified in the class fields only need to use the first unique set of characters of the category descriptions that have been previously defined in a CLASS data set.

Items

```
; Id (drawing no.)     Description     Configuration     Effectivity
:Description, Quantity, Weight, Xcg, Ycg, Zcg, Ixx, Iyy, Izz, Ixy, Ixz, Iyz, Overtyped, Material,
Shape, Volume Mode, Dim1, Dim2, Dim3, Dim4, Point1-X, Point1-Y, Point1-Z, Point2-X,
Point2-Y, Point2-Z, Point3-X, Point3-Y, Point3-Z, About-Z, About-Y, About-X

MPEX          DATASET     ITEMS
```

Due to the number of fields associated with this data set, which causes them to extend beyond the width of this page if placed on a single line as they might appear in a comment line, they are simply listed above in the order that they would appear.

The ITEMS data set uses the group relationship support of the MPEX format to define the details that make up a part. As such, the group definition specifies a particular part by giving its ID, Description, Configuration and Effectivity. The following lines of data, that are members of the group as indicated by the indentation character, are the items that make up the part. The items use a totally different set of fields to specify their characteristics. These fields include a description (which should be unique for each item within a part), quantity, the full mass properties, an overtyped field and the fields used to specify a shape, its dimensions and location. The overtyped field works just like it was described for the ENTRIES data set except in this case it specifies whether the item mass properties are calculated or manually defined. Calculated values have an overtyped value of 0 (false) and manually adjusted values have a value of 1 (true).

The shape is the description of the shape used to calculate the mass properties. The dimensions are given as Dim1 through Dim4. The orientation of the shape is specified with the three points where point 1 is the origin, point 2 is a point along the x-axis and point 3 is a point along the y-axis of the shape. A second method for orienting the shape is to use point 1 and the rotation angles. The rotation angles work the same as discussed in the ASSEMBLIES data set where the first rotation is about the z-axis of the part coordinate's system. The second rotation is about the rotated y-axis and the third rotation is the rotation about the resulting x-axis.

One needs to know how each shape is defined to fully understand how these dimensions and points define the size and orientation of the shape. Generally speaking, Dim1 gives the dimension along the x-axis, Dim2 gives the dimension along the y-axis and Dim3 gives the dimension along the z-axis. Dim4 usually is used to give the wall thickness of the shape.

Weight Changes

```
; Id (drawing no.)    Description     Configuration     Effectivity
; :Date    Delta Wt.    Result Wt.     Change Category     Change Reason       User
MPEX           DATASET      WEIGHT CHANGES
```

The WEIGHT CHANGES data set uses the group relationship support of the MPEX format to define the weight change history for a given part. As such, the group definition specifies a particular part by giving its ID, Description, Configuration and Effectivity. The following lines of data, that are members of the group as indicated by the indentation character, are the weight changes recorded for the part. The weight change data uses a totally different set of fields to specify it. These fields include a date, delta weight, the resulting weight, the weight change category associated with the change, a descriptive reason for the change and the user id of the person who made the weight change.

This data set does not use any parameters.

Notes

```
; Id (drawing no.)    Description     Configuration     Effectivity    Note
MPEX           DATASET      NOTES
```

The NOTES data set uses the standard MPEX format to describe the notes associated with a given entry record. There are no parameters defined for this dataset. A particular entry record identified by its ID, Description, Configuration and Effectivity has its note given as a long text field.

MPEX Example Usage

An example MPEX formatted file is available to go along with this document. If you did not receive it, contact the author by e-mail and he will forward it to you. The file is called MPEXAMPL.TXT. This file is also available at http://dbmass.com/mpex/mpexampl.txt.

Use the example to explore how to work with or create an MPEX file format. Open it using your favorite spreadsheet application (e.g. Excel). The Excel Text Import Wizard will recognize that it is a tab-delimited file. In this case you can simply select the Finish button to complete the import process. Save this file as a spreadsheet formatted file once you've imported it correctly. Then you can refer to it in the future without having to go through the importing text data process. Just remember, once the data is in the MPEX format within a spreadsheet or word processor application, save it as tab delimited text file to have it read by an MPEX translator.

Viewing MPEX Data

Since the MPEX formatted data is a text file, any text editor or spreadsheet application can be used to view, create or modify the data. When opening an MPEX file with a spreadsheet application you may need to specify that some columns of data are to be interpreted as text. This becomes necessary if leading spaces, periods or zeros appear in the column of data. This information may be intended as part of an Id or description, or used to indicate parent-child relationships. Excel’s “General” data format may drop these leading characters and/or convert the Id or description to numbers when they are intended to be interpreted as text. Other characters that can present a problem are plus, minus and equals (+, -, and =). When these characters appear at the beginning of a description the “General” data format interprets the description as a label or formula. Set the column format to text to avoid this problem.

Developing an MPEX Format Translator

Here are a few pointers that will help you with the process of developing an MPEX Translator.

Data Set Identification

Check for the Data Set Designator at the beginning of each new line.

Record Data

Remember that each record is terminated with a linefeed character. Therefore, after reading the fields of a record, search for the next linefeed character to identify the end of the record. Some data sets have more fields than others and when a spreadsheet is saved as a text file, extra tab characters can appear after the last defined field, even though there are no additional defined fields.

Hierarchical Relationships

When interpreting hierarchical relationships, you will need to keep track of whether a particular entry has had its sub-relationships defined. Once defined, ignore any subsequent definitions. This way multiple instances of a parent can be given within the overall product structure, but only the first instance will be used for its definition. This becomes important when one is bringing in data that lists the full product structure, as is common in report listings, since the MPEX format is intended to support this type of data.

Entry Types

Note that the entry types are specified by the data set. The STANDARD PARTS data set defines all entries that are to be interpreted as standard parts. The entry type parameter of the ASSEMBLIES data set gives the entry type information only for the top-level entries defined within the data set. So when interpreting the ASSEMBLIES data set, if the entry type of a child has not been previously defined, assume it is a PART. If, later, the entry appears as a parent, convert it to an ASSEMBLY. Or, if it is a top-level parent, use the entry type specified for the ASSEMBLIES data set. When no entry type parameter is specified for the ASSEMBLIES data set, assume the top-level parents are ASSEMBLIES.