**App Profile Creation Guide**

**Summary of document:** This document is to aid creation of App Profiles for JCAT. You should use other App Profiles for reference.

**Requirements/Assumptions for the XML:**

* Everything for the App is defined, and everything is in correct order, including
  + All <commands><command>s for the App
  + All <telemetry><parameter>s for the App
  + All <inputparameter>s and <choiceparameter>s in <commands><command><parameters> for the App
* The App’s commands must abide by the following requirement:
  + The command code for each command must be equal to <commandoffset> + X, where X is the index of the applicable <command> within <commands>

**Basic Format Example** (not a profile for any current App)**:**

<?xml version="1.0" encoding="UTF-8"?>

<channel>

<name>[APPNAME]</name>

<configs>

<config>

<name>[CPU 1]</name>

<cmdmid>[0x1804]</cmdmid>

<tlmmid>[0x0804]</tlmmid>

</config>

</configs>

<commandoffset>[+0]</commandoffset>

<commands></commands>

<telemetry></telemetry>

</channel>

**Notes:** The above example meets the minimum requirements to define an App Profile. The parts contained in brackets are specific to the App.

**New Terms:**

* <name>: The string to display as its name in the GUI
* <configs>: Contains the different configurations of the App.
* <config>: Contains a <name>, <cmdmid>, and <tlmmid>. You must define at least one <config> to load an App, as a config is required to send commands or receive telemetry. You can interact with more than one instance of an App by selecting a <config> for each instance at runtime.
* <commandoffset>: The lowest functional command code for the App. Usually 0.
* <commands>: Contains the various <command>s for an App.
* <telemetry>: Contains the various <parameter>s for an App Housekeeping Packet.

**Moderate-Depth Format Example** (the above example, with new parts **bolded)**

<?xml version="1.0" encoding="UTF-8"?>

<channel>

<name>[CFE\_TBL]</name>

<configs>

<config>

<name>[CPU 1]</name>

<cmdmid>[0x181D]</cmdmid>

<tlmmid>[0x081A]</tlmmid>

</config>

</configs>

<commandoffset>[+0]</commandoffset>

<commands>

**<command>**

**<name>CFE\_TBL\_VALIDATE\_CC</name>**

**<parameters>**

**<choiceparameter></choiceparameter>**

**<inputparameter></inputparameter>**

**<spare></spare>**

**</parameters>**

**</command>**

</commands>

<telemetry>

**<parameter>**

**<name>LastFileDumped</name>**

**<type>char</type>**

**<primitive>string</primitive>**

**<const>OS\_MAX\_PATH\_LEN</const>**

**</parameter>**

**<parameter>**

**<name>ByteAlignPad2</name>**

**<type>uint16</type>**

**</parameter>**

</telemetry>

</channel>

**Notes:** The above example is **not** a correctly-formatted App Profile. This is because, and only because, the contents within <choiceparameter>, <inputparameter>, and <spare> are not defined.

**New Terms:**

* <commands><command>: Defines a command that can be used on the App.
  + <name>: The name of the command.
  + <parameters>: The parameters for the command. This can be absent if there are no parameters for the command.
    - <choiceparameter>: A parameter where the user chooses from predetermined values in drop-down form.
    - <inputparameter>: A parameter where the user manually types in the value.
    - <spare>: A parameter that is invisible to the user, which contributes only byte values of zero to the packet.
* <telemetry><parameter>: Defines an aspect of telemetry about the App
  + <name>: The name of the parameter to display.
  + <type>: The data type of the parameter. The only valid values are:
    - int8, int16, int32
    - uint8, uint16, uint32
    - char
  + <primitive>: Defines whether or not the parameter is a string. If it is defined as “<primitive>string</primitive>”, it would be a string. If not, it is assumed an integer. If <primitive> is missing entirely, it is assumed an integer.
  + <const>: A variable whose value, as defined in the Constant Definition file, is the length of the array of <type>s in the string. This is used for command byte packing.

**In-Depth Format Example** (the above example, with new parts **bolded)**

<?xml version="1.0" encoding="UTF-8"?>

<channel>

<name>[CFE\_TBL]</name>

<configs>

<config>

<name>[CPU 1]</name>

<cmdmid>[0x181D]</cmdmid>

<tlmmid>[0x081A]</tlmmid>

</config>

</configs>

<commandoffset>[+0]</commandoffset>

<commands>

<command>

<name>CFE\_TBL\_VALIDATE\_CC</name>

<parameters>

<choiceparameter>

**<name>ActiveTblFlag</name>**

**<type>uint16</type>**

**<choice>**

**<name>CFE\_TBL\_INACTIVE\_BUFFER</name>**

**<value>0</value>**

**</choice>**

**<choice>**

**<name>CFE\_TBL\_ACTIVE\_BUFFER</name>**

**<value>1</value>**

**</choice>**

</choiceparameter>

<inputparameter>

**<name>TableName</name>**

**<type>char</type>**

**<const>CFE\_TBL\_MAX\_FULL\_NAME\_LEN</const>**

**<primitive>string</primitive>**

</inputparameter>

<spare>

**<type>uint8</type>**

</spare>

</parameters>

</command>

</commands>

<telemetry>

<parameter>

<name>LastFileDumped</name>

<type>char</type>

<primitive>string</primitive>

<const>OS\_MAX\_PATH\_LEN</const>

</parameter>

<parameter>

<name>ByteAlignPad2</name>

<type>uint16</type>

</parameter>

</telemetry>

</channel>

**Notes:** The above example **is** a correctly-formatted App Profile.

**New Terms:**

* <choiceparameter>
  + <name>: Defines the name of the <choiceparameter>.
  + <type>: The data type of the <choiceparameter>. The only valid values are:
    - int8, int16, int32
    - uint8, uint16, uint32
    - char
  + <choice>: Defines a selectable option from the GUI
    - <name>: The name to display as an option.
    - <value>: The integer value of the option, to be packed in the command packet as a uint8 integer.
* <inputparameter>
  + <name>: Defines the name of the <inputparameter>.
  + <type>: The data type of the <inputparameter>. The only valid values are:
    - int8, int16, int32
    - uint8, uint16, uint32
    - char
  + <const>: A variable whose value, as defined in the Constant Definition file, is the length of the array of <type>s in the string. This is used for command byte packing.
  + <primitive>: Defines whether or not the parameter is a string. If it is defined as “<primitive>string</primitive>”, it would be a string. If not, it is assumed an integer. If <primitive> is missing entirely, it is assumed an integer.
* <spare>
  + <type>: The data type of the <spare>. The only valid values are:
    - int8, int16, int32
    - uint8, uint16, uint32
    - char

Note: For a <spare>, the <type> is only used to denote the amount of byte to be allocated to it