Lesson 2

Objectives

- Introduce app specifications and how they are used by Basecamp

Notes

- This tutorial addresses building and running the Basecamp cFS target, so it describes how the app specs are used within that context. The Basecamp Application Developer's Guide goes into much more detail from a developer's perspective.
- The Raspberry Pi GPIO demo app (GPIO_DEMO) is used as an example. The complete app is located at https://github.com/cfs-apps/gpio_demo

App Specification

- **An app's complete specification for being integrated into the Basecamp cFS build and runtime systems is contained in 3 files**

1. **App Integration Spec, *[app].json***
   - A JSON file stored in the app's top-level directory that defines parameters for how to integrate an app with Basecamp's cFS target

2. **App Initialization Table, *[app]_ini.json***
   - A JSON file that is processed during an app's initialization that defines an app's runtime configuration parameters
   - In traditional NASA cFS apps, many of these parameters are defined in "mission config" and "platform config" header files that are defined when an app is compiled

3. **App Electronic Data Sheet, *[app].xml***
   - Defines an app's external interface

App Integration Spec

"app" defines the logistical parameters for managing the app

"cfs" defines the technical parameters for integrating an app into cFS target

"requires" defines an app's dependencies within a cFS target/system

```
{
   "app": {

   "title": "GPIO_DEMO",
   "version": "1.0.0",
   "supplier": "Open STEMware",
   "copyright": "bitValence",
   "url": "https://github.com/cfs-apps/gpio_demo",

   "description": [
      "This serves as a Raspberry Pi demo app. It use the GPIO interface to turn",
      "and off an LED."
   ],

   "cfs": {
      "cfe-type": "CFE_APP",
      "obj-file": "gpio_demo",
      "entry-symbol": "GPIO_DEMO_AppMain",
      "name": "GPIO_DEMO",
      "priority": 70,
      "stack": 32768,
      "load_addr": 0,
      "exception-action": 0,
      "app-framework": "osk",
      "tables": ["gpio_demo_ini.json"]
   },

   "requires": ["app_c_fw", "pi_iolib"]

}}
```

App Initialization Table

"config" defines an app's runtime configuration parameters that are a mixture of

- External interfaces
- Target-scope unique identifiers
- System resources
- Internal app-specific configurations

```
{
   "title": "Pi-Sat GPIO Demo initialization file",
   "description": [ "Define runtime configurations",
                    "GPIO Pin is the GPIO definition",
                    "and not the physical pin number"],

   "config": {

      "APP_CFE_NAME": "GPIO_DEMO",
      "APP_PERF_ID":  128,

      "APP_CMD_PIPE_NAME":  "GPIO_DEMO_CMD",
      "APP_CMD_PIPE_DEPTH": 10,

      "GPIO_DEMO_CMD_TOPICID"    : 0,
      "BC_SCH_4_SEC_TOPICID"     : 0,
      "GPIO_DEMO_HK_TLM_TOPICID" : 0,

      "CHILD_NAME":       "GPIO_DEMO_CHILD",
      "CHILD_PERF_ID":    44,
      "CHILD_STACK_SIZE": 16384,
      "CHILD_PRIORITY":   80,

      "CTRL_OUT_PIN" :   18,
      "CTRL_ON_TIME" : 3000,
      "CTRL_OFF_TIME": 6000
   }
}
```

Topic ID values are populated when '***make topicids***' is run

App Electronic Data Sheet

- **Electronic Data Sheets contain the definitions required to specify an app's external interface**
- **The *gpio_demo.xml* excerpt below shows the *RequiredInterfaceSet* and the *Implementation* of that interface**
- **The highlighted command definition fields illustrate how an interface definition is implemented**
  - The ParameterMap CMD interface has a VariableRef CmdTopicId that has an initial value that is defined in the CFE_MISSION package
- **Each GPIO_DEMO command (not shown) is defined as having a type "CommandBase"**

```
<RequiredInterfaceSet>
    <Interface name="CMD" shortDescription="Software bus telecommand interface" type="CFE_SB/Telecommand">
        <GenericTypeMapSet>
            <GenericTypeMap name="TelecommandDataType" type="CommandBase" />
        </GenericTypeMapSet>
    </Interface>
    <Interface name="HK_TLM" shortDescription="Software bus housekeeping telemetry interface" type="CFE_SB/Telemetry">
        <GenericTypeMapSet>
            <GenericTypeMap name="TelemetryDataType" type="HkTlm" />
        </GenericTypeMapSet>
    </Interface>
</RequiredInterfaceSet>
<Implementation>
    <VariableSet>
        <Variable type="BASE_TYPES/uint16" readOnly="true" name="CmdTopicId"    initialValue="${CFE_MISSION/GPIO_DEMO_CMD_TOPICID}" />
        <Variable type="BASE_TYPES/uint16" readOnly="true" name="HkTlmTopicId" initialValue="${CFE_MISSION/GPIO_DEMO_HK_TLM_TOPICID}" />
    </VariableSet>
    <!-- Assign fixed numbers to the "TopicId" parameter of each interface -->
    <ParameterMapSet>
        <ParameterMap interface="CMD"      parameter="TopicId" variableRef="CmdTopicId" />
        <ParameterMap interface="HK_TLM"  parameter="TopicId" variableRef="HkTlmTopicId" />
    </ParameterMapSet>
</Implementation>
```

Basecamp cfe-topicids.xml

- **cfe-eds-framework/basecamp_defs/eds/cfe-topicids.xml defines all of the Basecamp EDS Topic IDs in a package named CFE_MISSION**
  - GPIO_DEMO's EDS referenced the command topic ID with the following statement: initialValue="${CFE_MISSION/GPIO_DEMO_CMD_TOPICID}"
- **cfe-topicids.xml is organized in two groups: *telecommand* and *telemetry***
- **The telecommand and telemetry groups each contain a spare Topic ID list that are used for user apps**
- **The telecommand group contains a section that defines topics that are periodically sent by Basecamp's scheduling app.**
  - Apps can subscribe to receive these messages to perform periodic functions
  - Apps include scheduling topic IDs in their JSON initialization table
  - GPIO_DEMO's initialization table includes BC_SCH_4_SEC_TOPICID that is defined in cfe-topicids.xml as:

    <Define name="BC_SCH_4_SEC_TOPICID" value="${CFE_MISSION/TELECOMMAND_BASE_TOPICID} + 19"/>

Topic ID Integration Process

1. **Create app***
   a. Define topic IDs in app's EDS.xml file
   b. List Topic IDs used by the app in the app's JSON initialization table

2. **Add app to Basecamp****
   a. Add app's topic IDs to cfe-topicids.xml
   b. Add app's telemetry topic IDs to Basecamps Telemetry Output table

3. **Resolve app's topic IDs and build the Basecamp cFS target****
   a. Run 'make topicid'

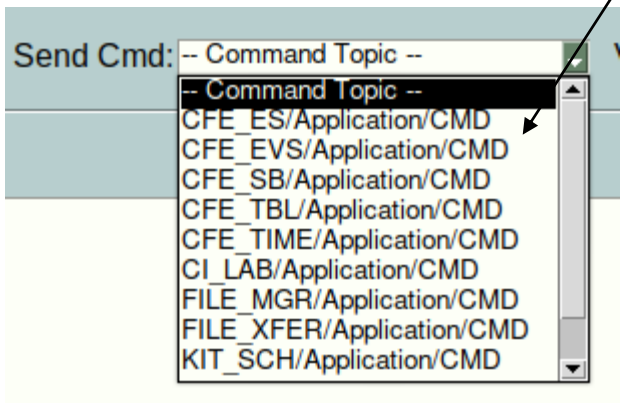
* Basecamp's 'Create App' GUI automatically performs these steps. The Basecamp Application Developer's Guide explains how to manually code it.

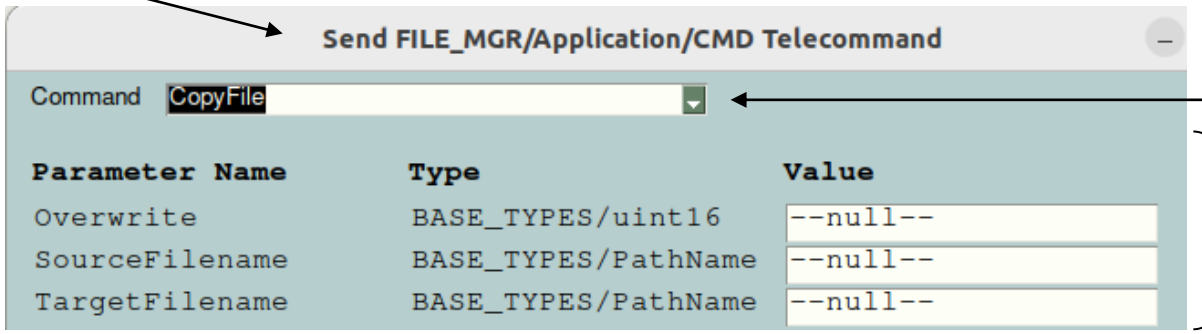
** Basecamp's 'Add App' GUI provides buttons to automatically peform these steps

Topic ID Runtime Usage

GUI telecommand and telemetry text is from Electronic Data Sheet (EDS) definitions

Telecommand Topics

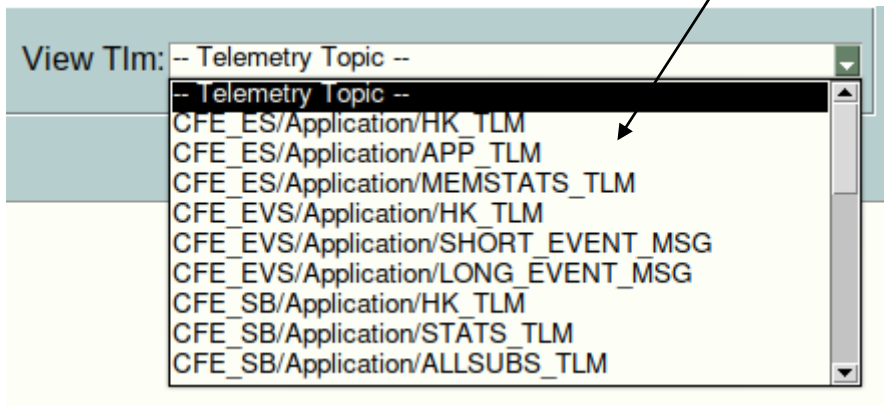


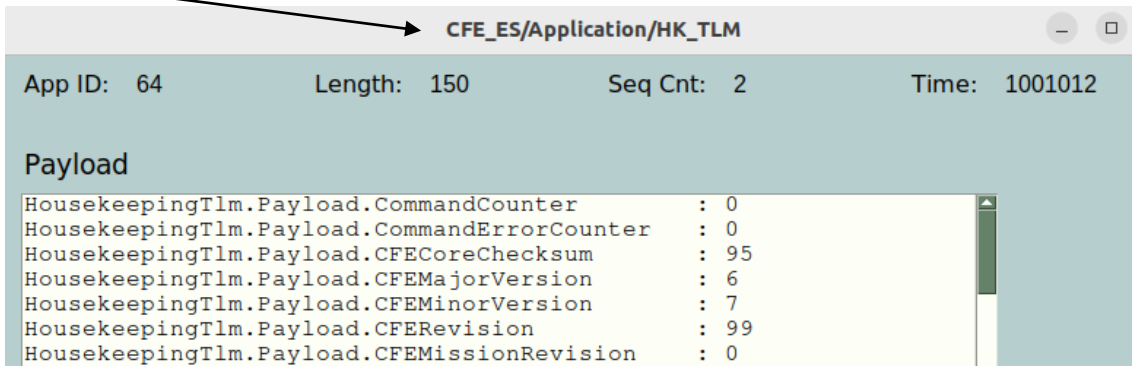


Command Name

Command Parameters

Telemetry topics





Telemetry Data Items

Electronic Data Sheet Artifact Summary

Inputs

cfe/fsw/cfe-core/eds

- base_types.xml
- ccsds_spacepackets.xml
- config.xml
- cFE_*[service]*.xml

***basecamp*_defs**

- [project]_mission_cfg.h
- [project]_perfids.h
- [target]_platform_cfg.h

***[project]*_defs/eds**

- mission_cfg.xml
- cfe_perfids.xml
- cfe_topicids.xml

apps/*[app]*/eds

- *[app]*.xml

FSW Outputs

build/inc

- base_types*.h
- ccsds*.h
- cfecfs_build_info.h
- cfecfs_version_info.h
- samplemission_eds_designparameters.h
- samplemission_eds_interfacedb.h
- samplemission_eds_master_index.h
- samplemission_eds_tgt_names.inc
- *[app]*_eds_defines.h
- *[app]*_eds_dictionary.h
- *[app]*_eds_dispatcher.h
- *[app]*_eds_interface.h
- *[app]*_eds_typedefs.h

build/[target]/default_cpu1/inc

- cfe_platform_cfg.h
- osconfig.h

build/exe/lib

- samplemission_eds_db.so
- samplemission_eds_interafcesdb.so

build/exe/lib/python

- CFE_MissionLib.so
- EdsLib.so

build/exe/host

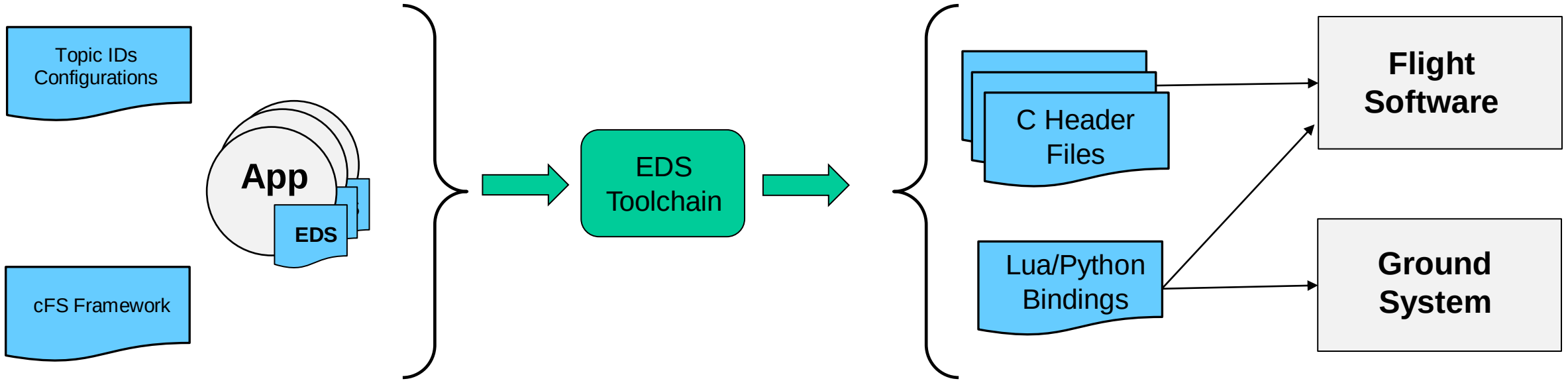
- Ground system tools

build/exe/cpu1

- FSW images, apps, etc.

[app] defines and typedefs do not depend on the cFS, they are pure C

Electronic Data Sheet Workflow



Single EDS definitions propagate to the ground and flight systems