



# cFS Basecamp Hello World Coding Lessons



**Version 1.10**  
**December 2023**

- These slides provide guidance for doing the Hello Object coding tutorial exercises
- The “Hello App Designs” section in Basecamp’s *Application Developer’s Guide* provides design information for all of Hello App coding tutorials
  - Having all of the design information in one place makes the developer’s guide flow better
  - It should be used in conjunction with this guide
- The Hello Object app template adds an example object to the Hello World application
  - The coding exercises introduce developer’s to the Basecamp’s object-based design strategy
- Prerequisites
  - Completed Hello World coding tutorial and met its prerequisites

## Commands

- Retain Hello World's Noop and Reset commands
- New Set Counter Mode command

Send HELLO\_OBJ/Application/CMD Telecommand

Command	Value
-- Command --	
Paramet	
Noop	
Reset	
SetCounterMode	

HELLO\_OBJ/Application/STATUS\_TLM - Port 9003

App ID: 116      Length: 13      Seq Cnt: 11

Payload

```
StatusTlm.Payload.ValidCmdCnt : 0
StatusTlm.Payload.InvalidCmdCnt: 0
StatusTlm.Payload.CounterMode : 1
StatusTlm.Payload.CounterValue : 43
```

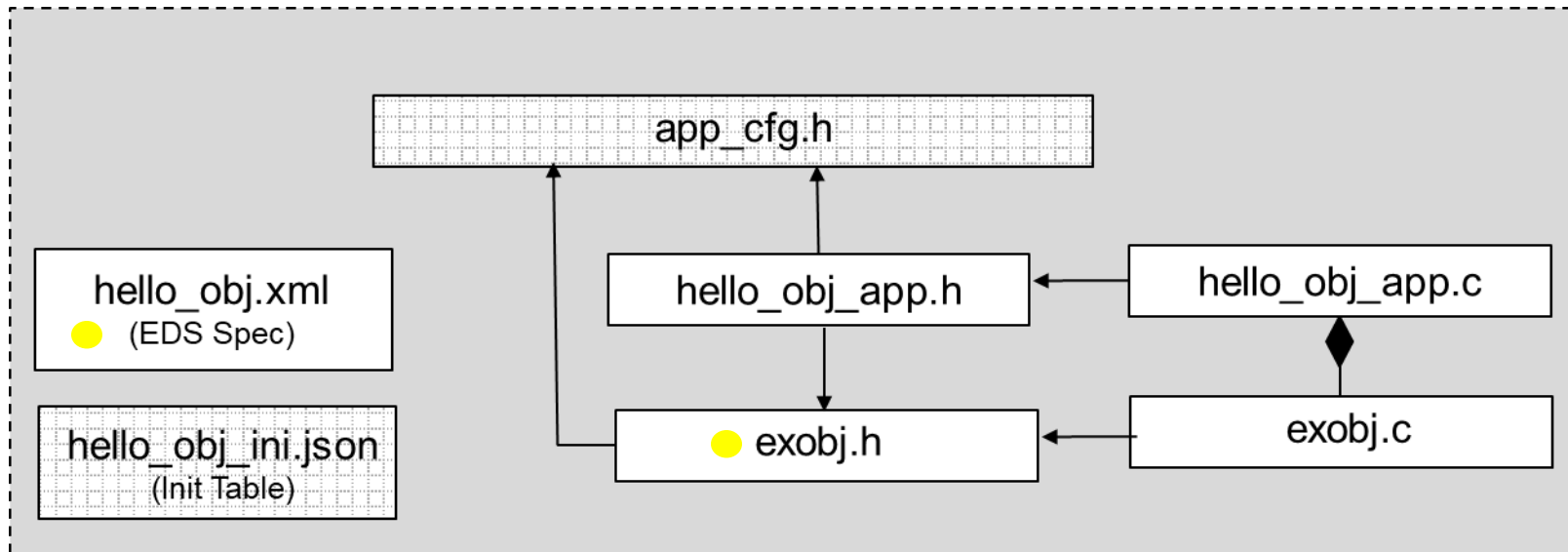
## Telemetry

- Retain Hello World command valid/invalid counters
- New Counter Mode data point
  - A code exercise changes the EDS definition so this will be a descriptive string
- New Counter Value data point
  - The counter updated at 1Hz and the status telemetry is sent every 4 seconds

## Objectives

- Increase knowledge of Electronic Data Sheets and how Basecamp apps use the code generated from the EDS

- The following files are modified in this lesson





## hello\_obj.xml

- CounterMode enumeration definition

```
<EnumeratedDataType name="CounterMode" shortDescription="" >
  <IntegerDataEncoding sizeInBits="16" encoding="unsigned" />
  <EnumerationList>
    <Enumeration label="Increment" value="1" shortDescription="" />
    <Enumeration label="Decrement" value="2" shortDescription="" />
  </EnumerationList>
</EnumeratedDataType>
```

← Zero is avoided unless the enumeration is used as an index into array

- The EDS toolchain created a file named hello\_obj\_eds\_typedefs.h
- The Basecamp app convention is to include the EDS typedefs in app\_cfg.h
  - Basecamp app designs are object-based, however EDS specs define app level interfaces that may include definitions from multiple objects so adhering to type definitions encapsulated within an object is not always achievable

## exobj.h

- The EDS toolchain creates the following variable for the enumeration definition: *HELLO\_OBJ\_CounterMode\_Enum\_t*
  - HELLO\_OBJ* is the package name
  - CounterMode* is the enumerated data type name
  - Enum\_t* is a naming convention used by the EDS toolchain

## Verification

1. Use the main screen's cFS Build button to build the target (only existing files changed)
2. Since the EDS was modified, the GUI ,must be restarted so the new telemetry type is used
3. After starting the cFS open the HELLO\_OBJ status telemetry page

- The default mode should be Increment ☐

**HELLO\_OBJ/Application/STATUS\_TLM - Po**

App ID: 116      Length: 13      Seq Cnt: 2

**Payload**

```
StatusTlm.Payload.ValidCmdCnt : 0
StatusTlm.Payload.InvalidCmdCnt: 0
StatusTlm.Payload.CounterMode : Increment
StatusTlm.Payload.CounterValue : 7
```

4. Issue a SetCounterMode decrement command

**Send HELLO\_OBJ/Application/CMD Telecommand**

Command: **SetCounterMode**

Parameter Name	Type	Value
Mode	HELLO_OBJ/CounterMod	Decrement



**HELLO\_OBJ/Application/STATUS**

App ID: 116      Length: 13      Seq C

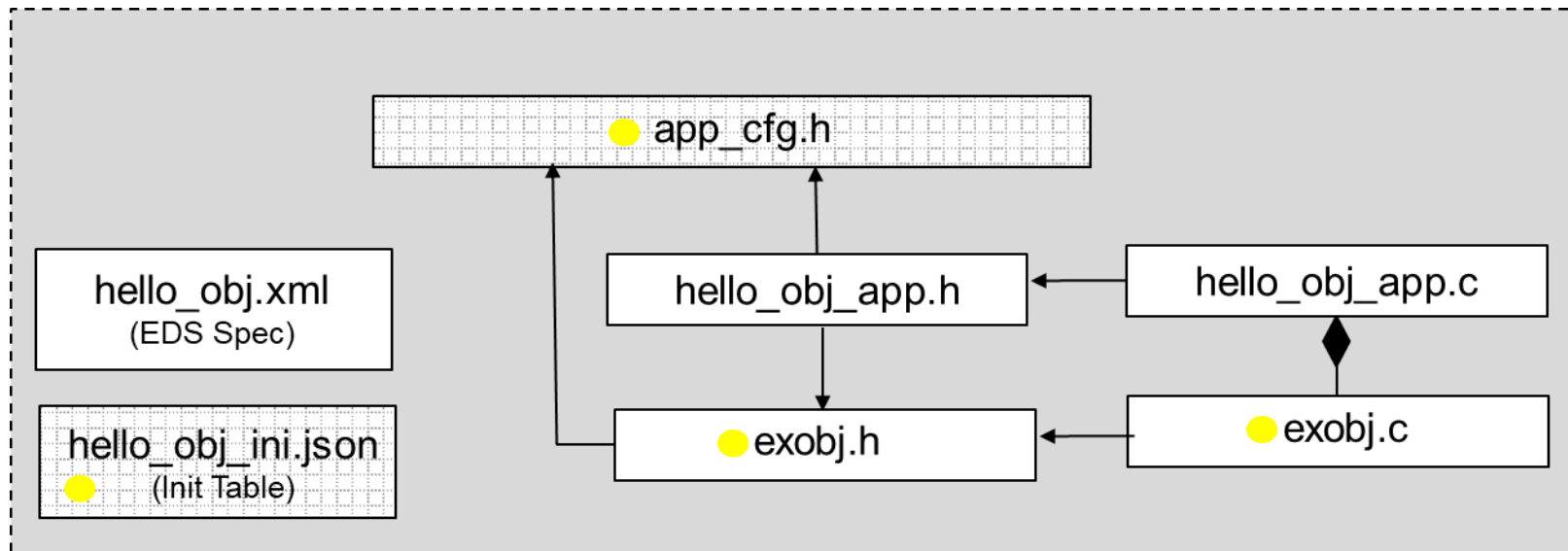
**Payload**

```
StatusTlm.Payload.ValidCmdCnt : 1
StatusTlm.Payload.InvalidCmdCnt: 0
StatusTlm.Payload.CounterMode : Decrement
StatusTlm.Payload.CounterValue : 100
```

## Objectives

- Learn app\_cfg.h architectural role
- Learn object-based strategy and role of encapsulation
- Build upon Hello World JSON init table lesson and learn init file app level management and relationship to objects

- The following files are modified in this lesson



## **cpu1\_hello\_obj\_ini.json**

- Defining limits in the init file means the limits can be changed between app restarts without the need for recompilation
- If you need the ability to change limit changes during runtime without restarting the app then either a command is needed or the limits can be defined in a table

## **app\_cfg.h**

- Add the limit parameter declarations

## **exobj.h**

- Remove the hardcoded limits and add class variables to store the limit values

## **exobj.c**

- Initialize the limits in the example object's constructor.
- If an init parameter is needed once and remains unchanged during runtime then it is used in the app's main init function or an object's constructor and it doesn't need to be stored in a variable. Otherwise it should be stored in a variable.
- Because of the example app's encapsulation, the main app files did not need modification for this lesson



## Verification

1. After the app starts observe new range taking effect
  - Here is the first packet after HELLO\_OBJ starts (sequence count equals 1) and the *Counter Value* is 13 which is slightly more than the low limit of 10
  - This is reasonable because the counter is incremented at 1Hz and the status telemetry is sent every 4 seconds

HELLO\_OBJ/Application/STATUS\_TLM - P

App ID: 116

Length: 13

Seq Cnt: 1

Payload

StatusTlm.Payload.ValidCmdCnt : 0

StatusTlm.Payload.InvalidCmdCnt: 0

StatusTlm.Payload.CounterMode : Increment

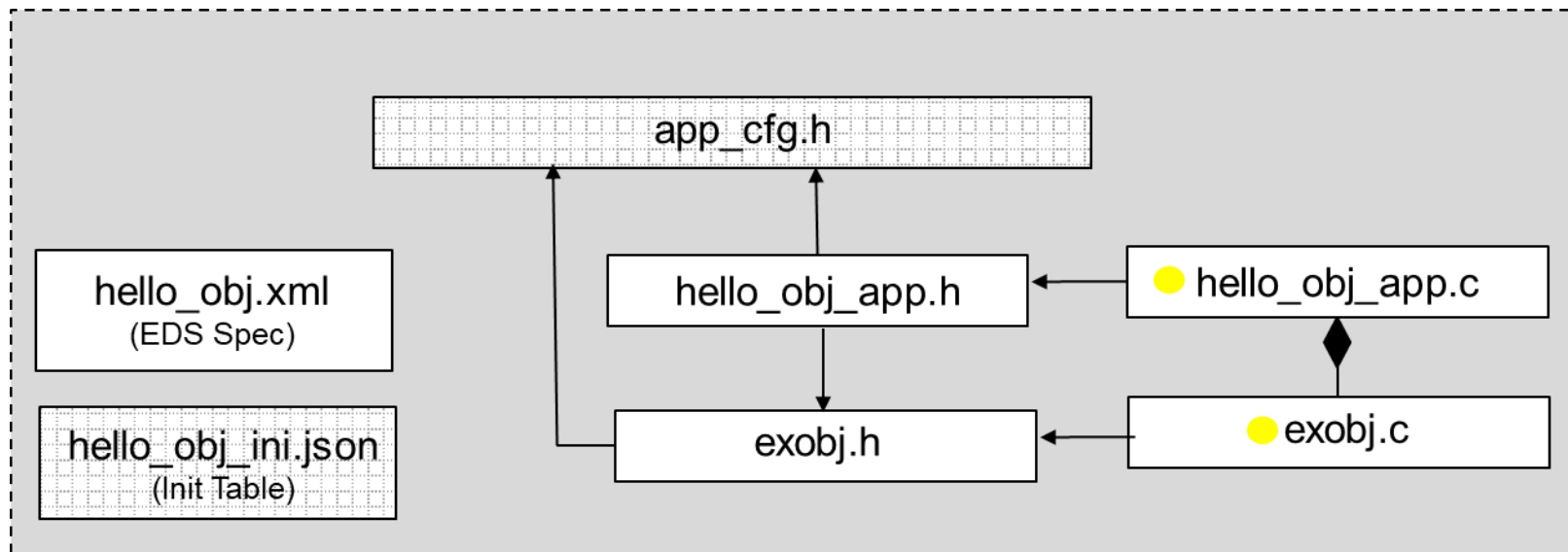
StatusTlm.Payload.CounterValue : 13

2. Change HELLO\_OBJ's init table limits and restart the app to observe the different limits taking effect
  - See the Demo App lesson in the Basecamp Introduction tutorial for guidance on how to perform these steps

## Objectives

- Introduce event types and event filters
- Use an object with a periodic function that is triggered by a Scheduler (SCH) app message
- Deeper dive into app architecture, resources, roles and responsibilities

- The following files are modified in this lesson



## hello\_obj.c

- The event message filter array is owned by the app's main object
- The event message IDs are defined by the objects owned by the app
- The cFE Event Service defines several macros such as CFE\_EVS\_FIRST\_8\_STOP which is used in this lesson
- It's up to the user whether to add the call to CFE\_EVS\_ResetAllFilters() as part of the app's reset command. The event services provides a commands to reset individual app event filters

## exobj.c

- The Execute()'s debug event message is changed to an INFORMATION type which causes it to be continuously sent until the filter limit is reached
- In general a type of DEBUG makes sense for this kind of event message and it was changed in this lesson to demonstrate event message filters and to demonstrate how event types can be used

## Verification

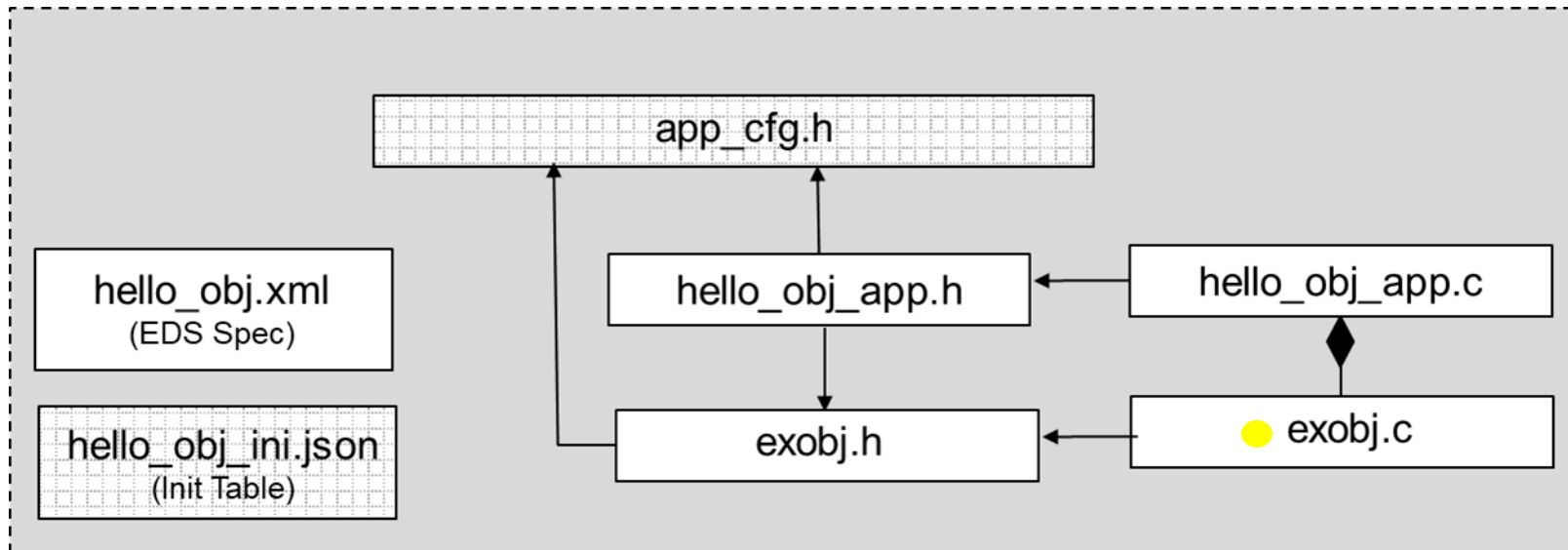
- Observe 8 INFORMATION event messages are sent when the app first starts running and then the filter limit is reached so the messages are no longer output
- When you issue an app reset command you should see 8 more messages get sent because the event filter is reset

Ground Events	Clear
17:40:20 - FSW Event at 1040864: HELLO_OBJ, 2 - INCREMENT counter mode: Value 15	
17:40:21 - FSW Event at 1040864: KIT_SCH, 2 - Multiple slots processed: slot = 1,	
17:40:21 - FSW Event at 1040864: HELLO_OBJ, 2 - INCREMENT counter mode: Value 16	
17:40:22 - FSW Event at 1040866: HELLO_OBJ, 2 - INCREMENT counter mode: Value 17	
17:40:23 - FSW Event at 1040866: KIT_SCH, 3 - Major Frame Sync too noisy (Slot 1).	
17:40:38 - Sent HELLO_OBJ/Reset command	
17:40:40 - FSW Event at 1040884: HELLO_OBJ, 2 - INCREMENT counter mode: Value 35	
17:40:41 - FSW Event at 1040885: HELLO_OBJ, 2 - INCREMENT counter mode: Value 36	
17:40:42 - FSW Event at 1040886: HELLO_OBJ, 2 - INCREMENT counter mode: Value 37	
17:40:43 - FSW Event at 1040887: HELLO_OBJ, 2 - INCREMENT counter mode: Value 38	
17:40:44 - FSW Event at 1040888: HELLO_OBJ, 2 - INCREMENT counter mode: Value 39	
17:40:45 - FSW Event at 1040889: HELLO_OBJ, 2 - INCREMENT counter mode: Value 40	
17:40:46 - FSW Event at 1040890: HELLO_OBJ, 2 - INCREMENT counter mode: Value 41	
17:40:47 - FSW Event at 1040891: HELLO_OBJ, 2 - INCREMENT counter mode: Value 42	

## Objectives

- Show how objects can have functional requirements for an App's Reset command
- Demonstrate how event message filters can be useful for debugging periodic events

- The following files are modified in this lesson





## exobj.c

- The Basecamp interpretation of the “standard” Reset app command functionality is broader than the cFS convention
- The cFS convention is to focus on counters and this grew out of the need to reset the valid/invalid command counters
- Basecamp apps reset state and it’s up to the mission/developer as to what makes sense
- Be careful not to make it reset too much when it would be beneficial to have individual commands to reset specific functionality
- EXOBJ\_Execute() runs at 1Hz so having the debug event filtered allows the developer to enable debug events to prove the function is running as expected and the events stop so telemetry is not cluttered
- Ultimately the event strategy should match the debugging scenario needs

## Verification

1. Reset the app and observe the counter go to the low limit
2. Change the counter mode to decrement, wait and then reset to show it is reset to the high limit (not shown)

HELLO\_OBJ/Application/STATUS\_TLM - P

App ID: 116

Length: 13

Seq Cnt: 21

Payload

Display Paused

StatusTlm.Payload.ValidCmdCnt : 1

StatusTlm.Payload.InvalidCmdCnt: 0

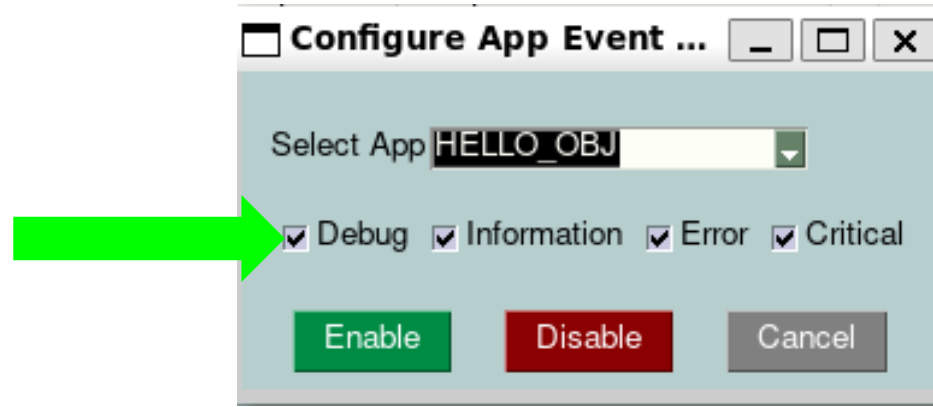
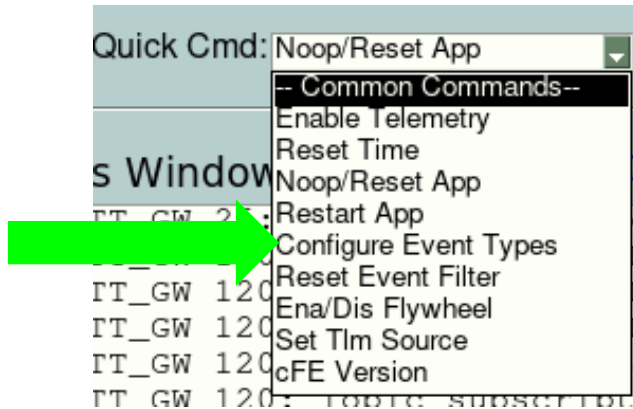
StatusTlm.Payload.CounterMode : Increment

StatusTlm.Payload.CounterValue : 13

- First telemetry message after reset command
- Display paused
- Counter value of 13 makes sense because near the low limit of 10 and telemetry is output every 4 seconds

## Verification Continued

3. Enable debug messages and see the execute debug events
4. Reset app and show the filter is reset and applies to any event types (not shown)



Use Configure Event command to enable HELLO\_OBJ's debug events

First 8 debug events are output

```
07:41:23 - Sent CFE_EVS/EnableAppEventTypeCmd command
07:41:25 - FSW Event at 1001297: HELLO_OBJ, 1 - INCREMENT counter mode: Value 22
07:41:26 - FSW Event at 1001298: HELLO_OBJ, 1 - INCREMENT counter mode: Value 23
07:41:27 - FSW Event at 1001299: HELLO_OBJ, 1 - INCREMENT counter mode: Value 24
07:41:28 - FSW Event at 1001300: HELLO_OBJ, 1 - INCREMENT counter mode: Value 25
07:41:29 - FSW Event at 1001301: HELLO_OBJ, 1 - INCREMENT counter mode: Value 26
07:41:30 - FSW Event at 1001302: HELLO_OBJ, 1 - INCREMENT counter mode: Value 27
07:41:31 - FSW Event at 1001303: HELLO_OBJ, 1 - INCREMENT counter mode: Value 28
07:41:32 - FSW Event at 1001304: HELLO_OBJ, 1 - INCREMENT counter mode: Value 29
```