cFS Basecamp
Hello Table Coding Lessons

Version 2.8

November 2025

Tutorial Introduction

- **These slides provide guidance for doing the Hello Table 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 Table app template adds an example table to the Hello Object application**
  - The coding exercises introduce developer's to the Basecamp's JSON table design strategy and operations

- **Prerequisites**
  - Completed Hello Object coding tutorial and met its prerequisites

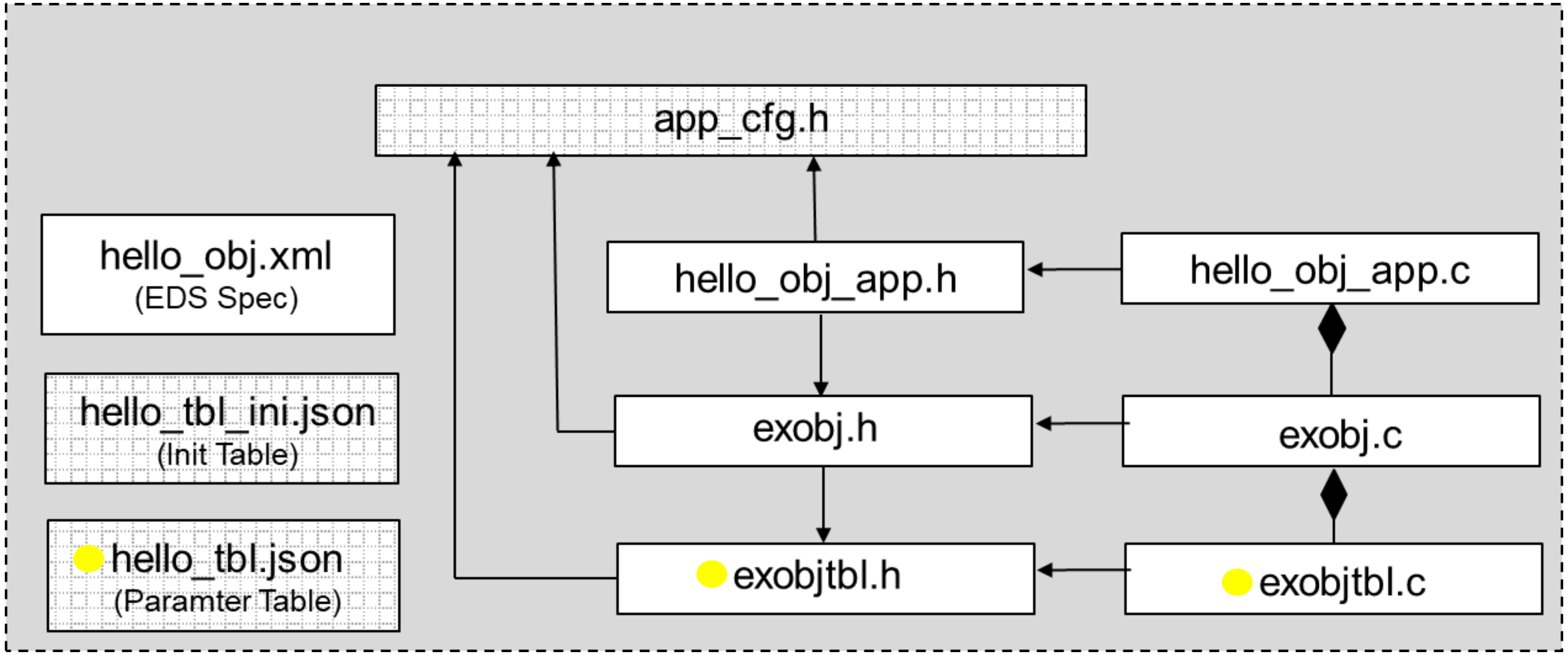
Lesson 1 – Add a New Table Parameter (1 of 2)

Objectives

- Learn how a JSON table parameter is stored, parsed during a load command, and written to a dump file

● **The following files are modified in this lesson**

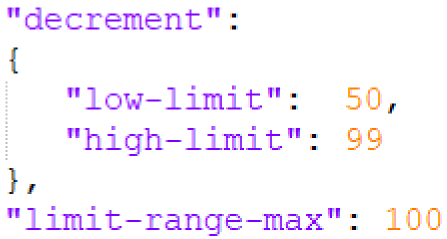
- Note how object encapsulation limits the change to the table files



Lesson 1 – Add a New Table Parameter (2 of 2)

hello_tbl.json

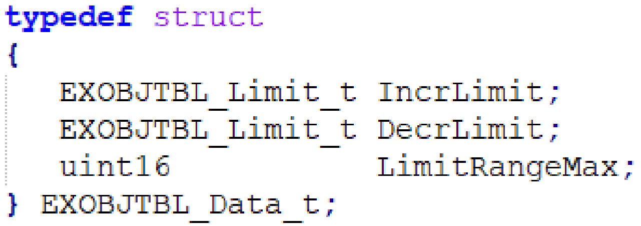
- Add a new parameter “limit-range-max” to the end of the table



```
"decrement":
{
   "low-limit":  50,
   "high-limit": 99
},
"limit-range-max": 100
```

exobjtbl.h

- Table objects provide local storage that is used during a table load
- After a table is loaded and optionally validated, the data is copied into the storage of the object that owns the table



```
typedef struct
{
   EXOBJTBL_Limit_t IncrLimit;
   EXOBJTBL_Limit_t DecrLimit;
   uint16           LimitRangeMax;
} EXOBJTBL_Data_t;
```

exobjtbl.c

- Table file contain the following declaration *static CJSON_Obj_t JsonTblObjs[]* that is used during a table load to instruct the JSON parser how to parse the JSON object and where to store the result
- Table dumps require the developer to hand code print statements to write the JSON objects to a file

Lesson 1 – Build New cFS Target

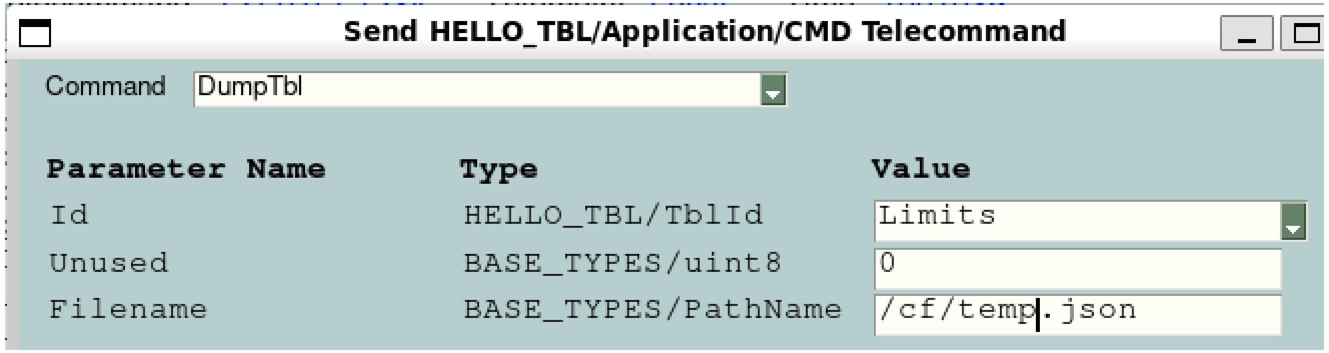
1. **Use the main screen's cFS Build button to build the target**
   - Only existing files changed, so no need to perform a Build New

2. **Since the EDS was not modified, the GUI does not need to be retarted**

3. **The following slides describe how to verify the new table parameter**

Lesson 1 – Verify New Table Parameter

1. **Issue a DumpTbl command to a filename other than the default table name**

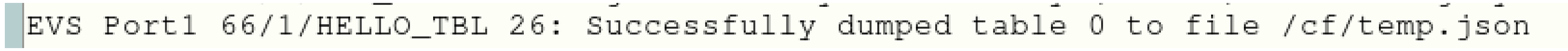


Send HELLO_TBL/Application/CMD Telecommand

Command: DumpTbl

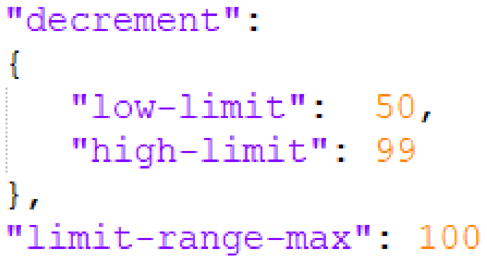
Parameter Name	Type	Value
Id	HELLO_TBL/TblId	Limits
Unused	BASE_TYPES/uint8	0
Filename	BASE_TYPES/PathName	/cf/temp.json

2. **You should see an event message indicating the table was written to the commanded filename**



```
EVS Port1 66/1/HELLO_TBL 26: Successfully dumped table 0 to file /cf/temp.json
```

3. **Transfer the file to the ground and open it with a text editor**
   - The values should be identical with the hello_tbl.json values



```
"decrement":
{
   "low-limit":  50,
   "high-limit": 99
},
"limit-range-max": 100
```

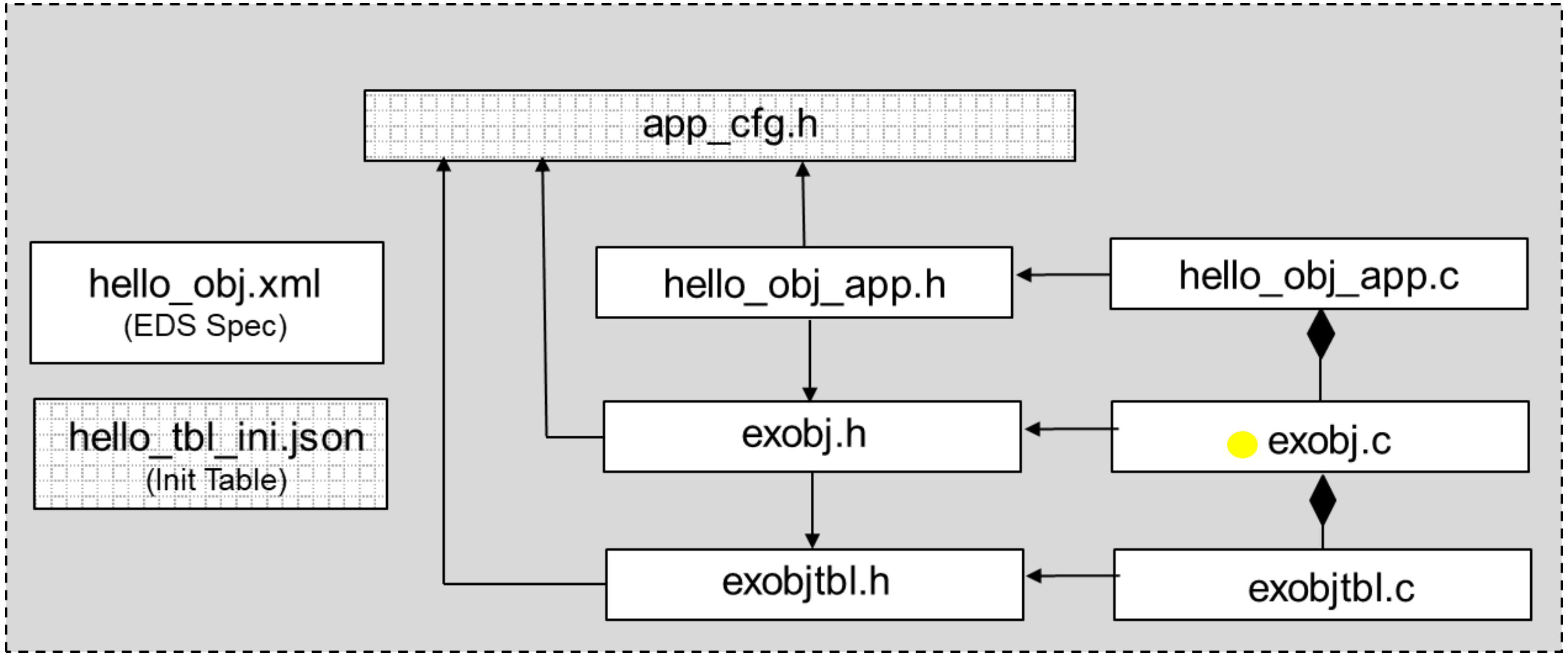
Lesson 2 – Add Table Load Acceptance Check (1 of 2)

Objectives

- Learn how to supply a table load validation function

● The following files are modified in this lesson

- Note only exobj needs to change because it has the contextual knowledge of how the table parameters are used



Lesson 2 – Add Table Load Acceptance Check (2 of 2)

exobj.c

- A pointer to the table validation function is passed to the EXOBJTBL’s constructor
  - EXOBJTBL_Constructor(&ExObj->Tbl, IniTbl, AcceptNewTable);
  - This function is called as part of the table load command

- The new validation logic ensures a the table range limits are within the maximum allowed

Lesson 1 – Build New cFS Target

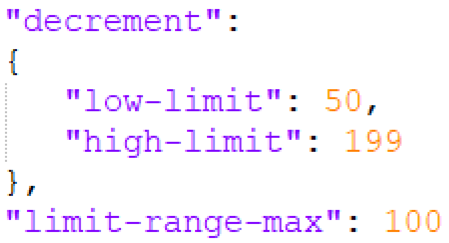
1. **Use the main screen's cFS Build button to build the target**
   - Only existing files changed, so no need to perform a Build New

2. **Since the EDS was not modified, the GUI does not need to be retarted**

3. **The following slides describe how to verify the new table parameter**

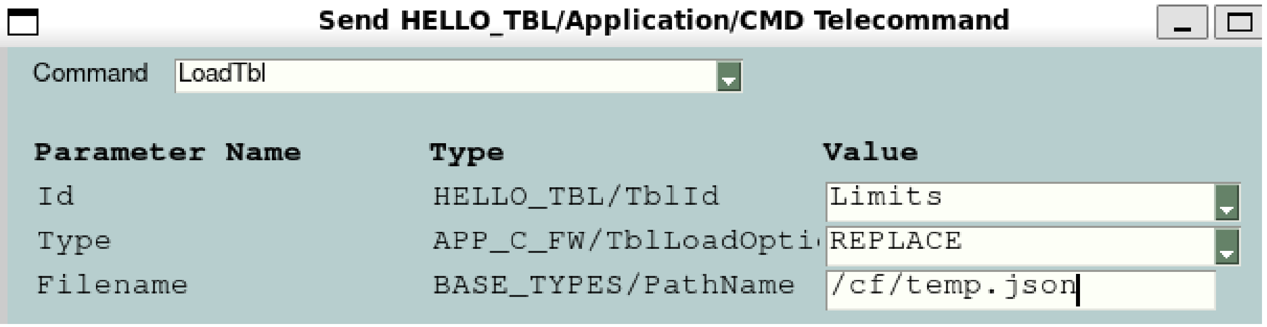
Lesson 2 – Verify Table Load Acceptance Check

1. **Edit the temp.json dump file created in lesson one with an invalid range such as this one**



```
"decrement":
{
   "low-limit": 50,
   "high-limit": 199
},
"limit-range-max": 100
```

2. **Issue a LoadTbl command**

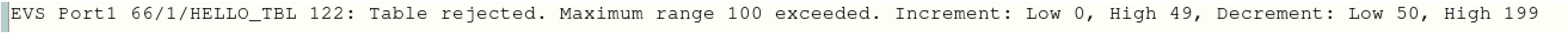


Send HELLO_TBL/Application/CMD Telecommand

Command: LoadTbl

Parameter Name	Type	Value
Id	HELLO_TBL/TblId	Limits
Type	APP_C_FW/TblLoadOpti	REPLACE
Filename	BASE_TYPES/PathName	/cf/temp.json

3. **You should see an error event message stating the table load failed**
   - Note the TBLMGR service incorrectly reports the last table load was valid. This bug is captured in issue #59



```
EVS Port1 66/1/HELLO_TBL 122: Table rejected. Maximum range 100 exceeded. Increment: Low 0, High 49, Decrement: Low 50, High 199
```