



“NASA Table” NASA Style App Coding Lessons



Basecamp Version 2.7
July 2025

Tutorial Introduction



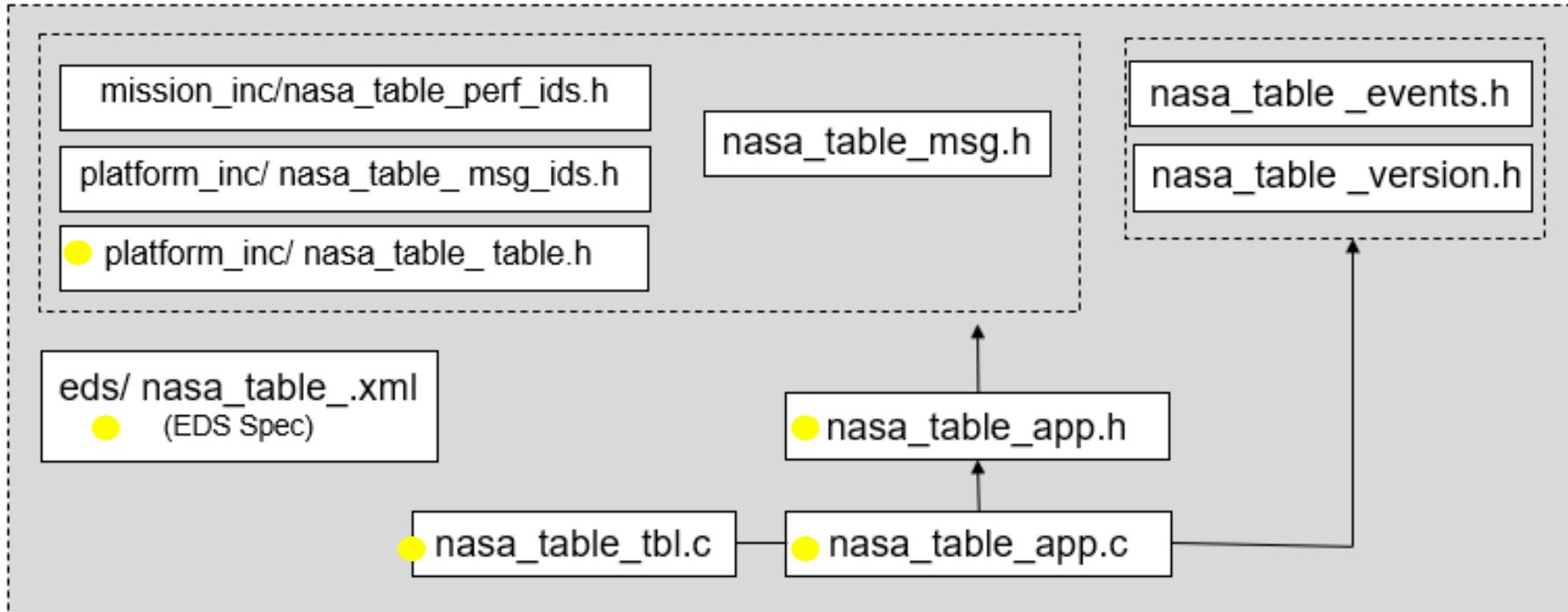
- These slides supplement the *NASA Table* coding tutorial exercises
- The “Hello App Designs” section in Basecamp’s *Application Developer’s Guide* provides design information for all of Hello App coding tutorials
 - Having all of the design information in one place makes the developer’s guide flow 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 developers to the Basecamp’s JSON table design strategy and operations
- Prerequisites
 - Completed *NASA World* coding tutorial and met its prerequisites

Lesson 1 – Add Table (1 of 3)

Objectives

- Learn how to add a table to an application
 - This lesson creates a default table that is built with the application
-
- The following files are modified in this lesson

App Source Files



CMakeLists.txt

- Add instructions so the table source file is included in the build and the table tool creation tool is invoked to create the .tbl file.

nasa_table.h

- Define table constants and the table data structure.

nasa_table_tbl.c

- Define default table values
- Use CFE_TBL_FILEDEF() to define the table for the table build tool

nasa_table_tbl.xml

- Define the table data structure and it must match definitions in nasa_table_tbl.c
- The need for this duplicate definition is transitory should be removed once the EDS toolchain is fully integrated in the NASA cFS public release

nasa_table_app.h

- Add a “table handle” variable to the app’s global data structure
- Table handles are created when a table is registered with the cFE and are used in subsequent table API calls to operate on a specific table
- Unlike Sample App that contains an array of table handles, this tutorial only has a single handle

nasa_table_app.c

- Include the table header file
- Define a static function prototype for a local file function that will report the table values in an event message
- In the app’s init function register the table with the cFE
- In the NOOP command function add a call to the static function that will report the table values
- Add the static function to report the table values

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 modified, the GUI must be restarted so the new EDS library with the new command is used**
- 3. The following slides describe how to use the new command**

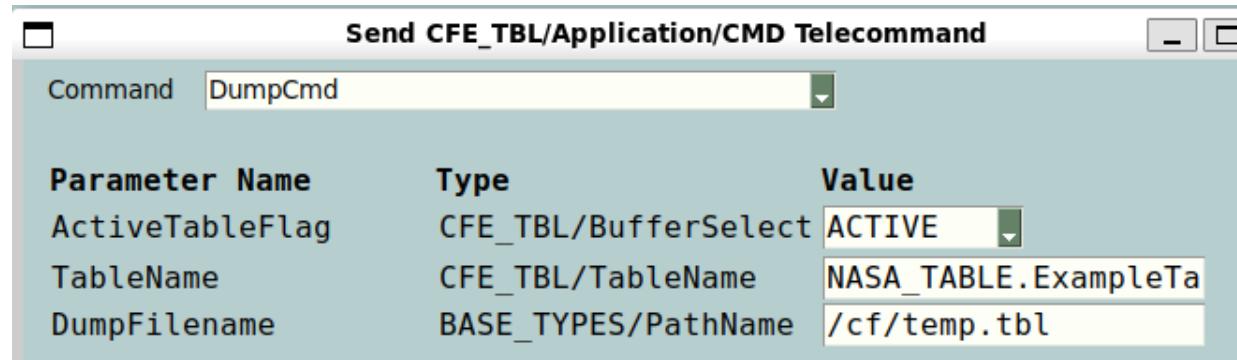
Lesson 1 – New Table Operations (1 of 3)



- Issue a NASA_TABLE NOOP command and you should see the default table values reported in an event message

```
EVS Port1 66/1/NASA_TABLE 3: NASA_TABLE: NOOP command v1.3.0-rc4+dev39  
1980-012-14:55:23.25069 NASA_TABLE: Table Value 1: 1  Value 2: 2
```

- Issue a cFE Table Dump command as follows:



- The following event message indicates the table was successfully written to a file

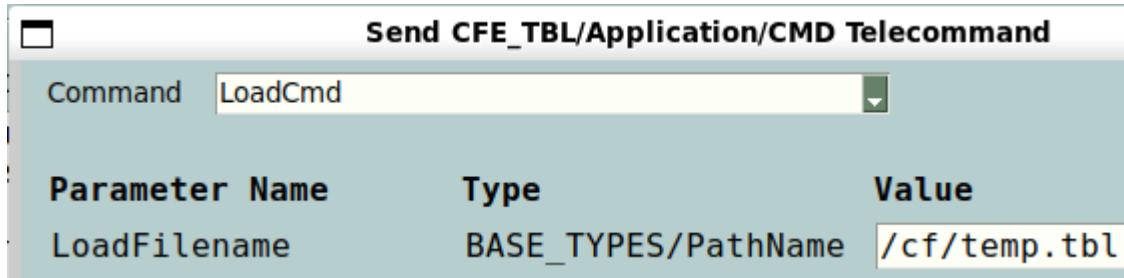
```
EVS Port1 66/1/CFE_TBL 14: Successfully dumped Table 'NASA_TABLE.ExampleTable' to '/cf/temp.tbl'
```

- The **Active Buffer** is the buffer currently being used by an application

Lesson 1 – New Table Operations (2 of 3)



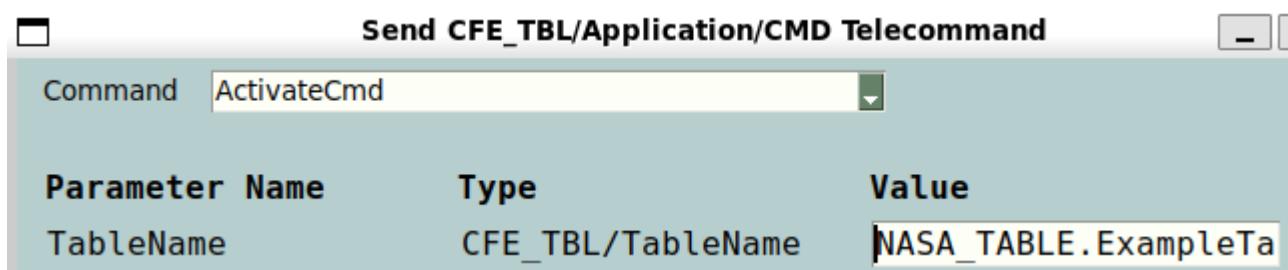
3. Issue a cFE Table Load command to load the /cf/temp.tbl



- The following event message indicates the table was successfully loaded

```
EVS Port1 66/1/CFE_TBL 12: Successful load of '/cf/temp.tbl' into 'NASA_TABLE.ExampleTable' working buffer
```

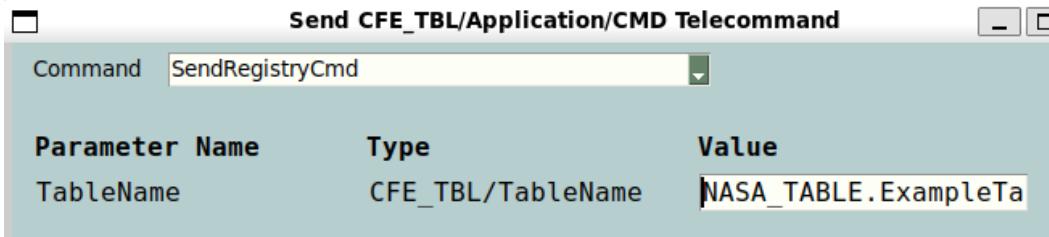
- This command loaded the table into a working buffer. You can try to activate the table but nothing will happen because we need to add code to manage table loads.



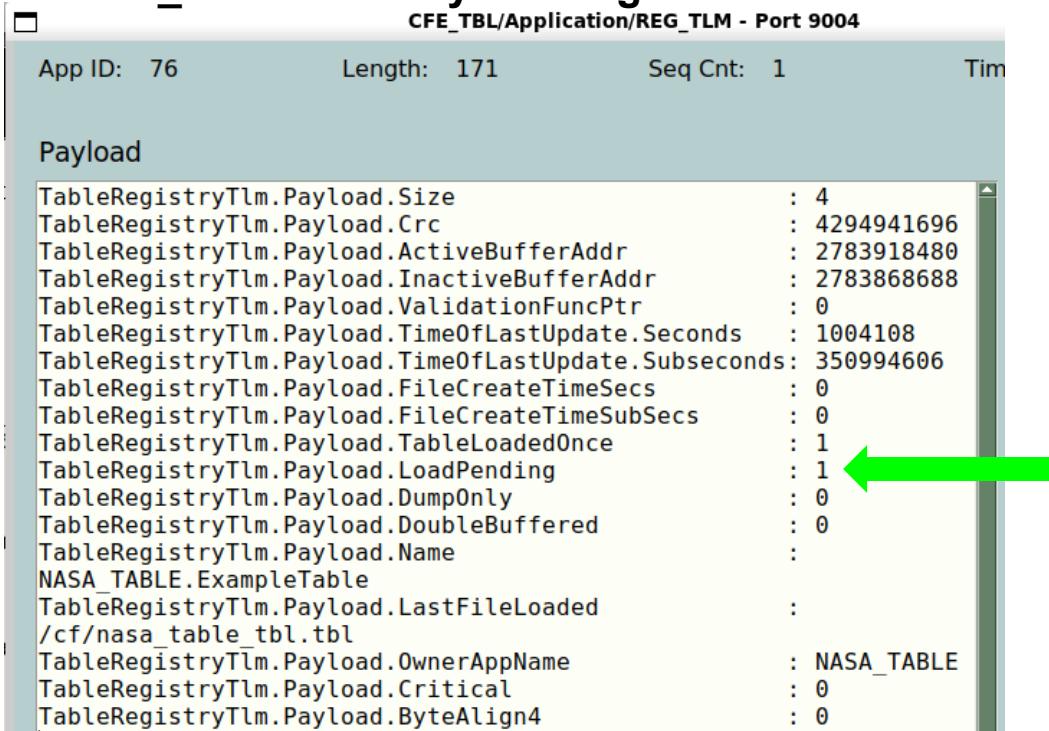
Lesson 1 – New Table Operations (3 of 3)



4. A cFE Table *Send Registry* command can be used to show a table load is pending for NASA_WORLD



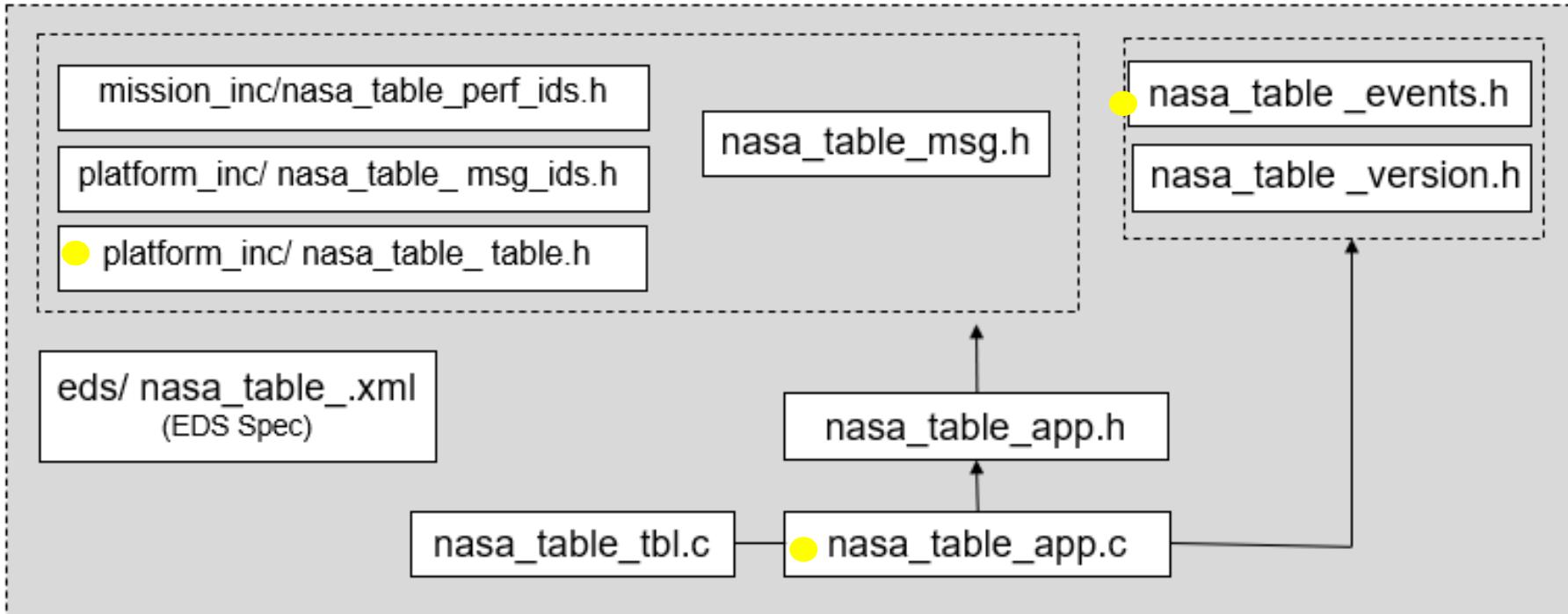
- This sends a *REG_TLM* telemetry message and the *LoadPending* is set



Objectives

- Learn how to use the 'housekeeping cycle' to manage table loads and how to provide a function that validates a table's content prior to it being activated
- The following files are modified in this lesson

App Source Files



nasa_table_table.h

- Define constants used by the table validation function:
 - A maximum positive numeric limit for the Int1 table element
 - An error to return if the validation fails

nasa_table_events.h

- Define constants used by the table validation function:

nasa_table_app.c

- Define a table validation function that verifies the Int1 table element is \leq to the maximum value constant
- Pass the address of the table validation function to CFE_TBL_Register()
- After the Housekeeping telemetry message is sent call CFE_TBL_Manage()
 - This checks to see whether a table load is pending
 - If there is a pending load the app's table validation function is called (if one provide during registration)
 - If validation passes then the table is activated

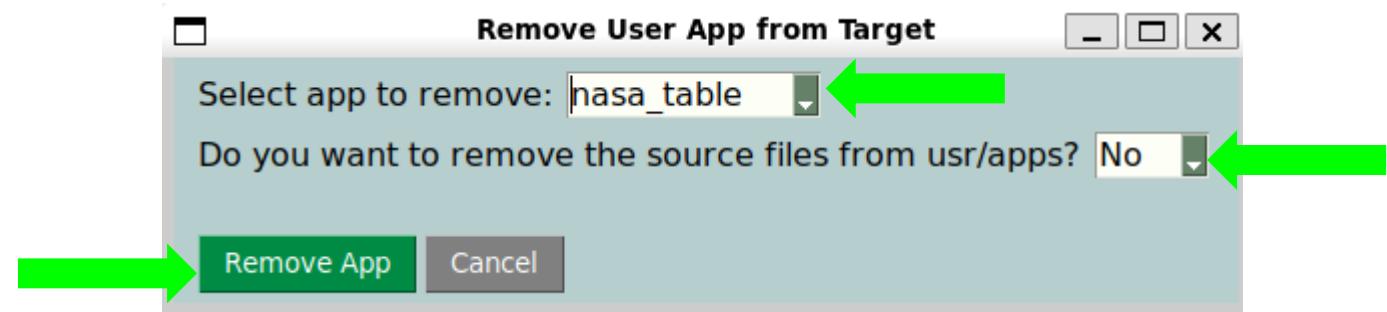
Lesson 2 – Build New cFS Target (1 of 2)



IMPORTANT

This lesson is unique because the NASA_TABLE app needs to be removed from the target and then added back to the target

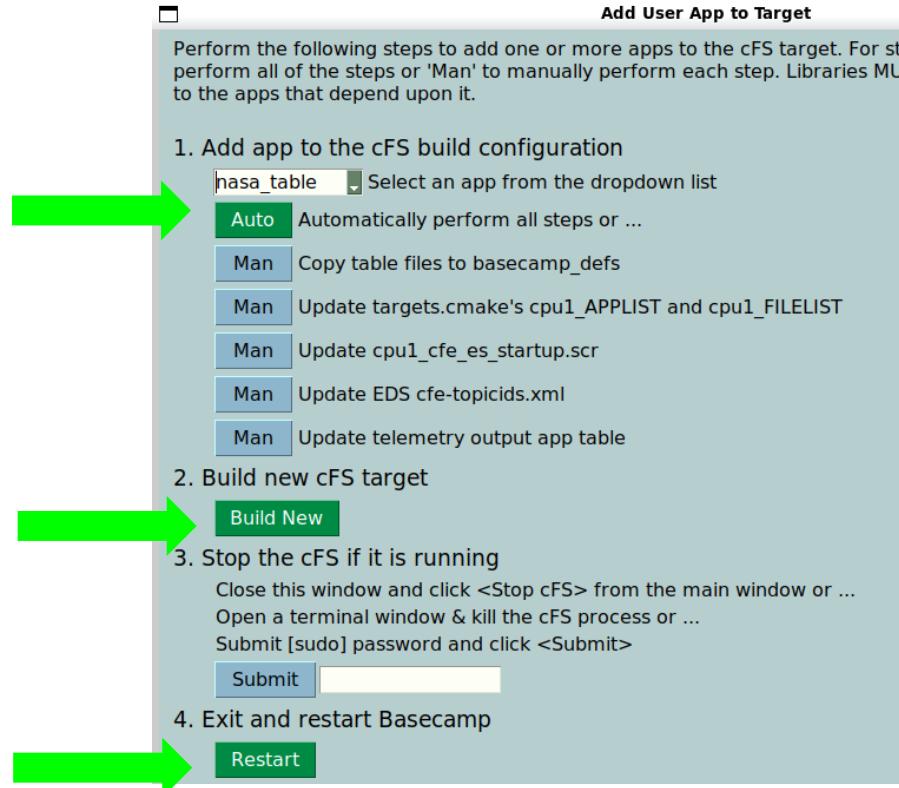
- **Reinstalling the app in the target is required because a new Topic ID was created for the table's proxy telemetry**
 - The application target installation process adds an app's Topic IDs to cfe-topicids.xml
1. **From Basecamp's main menu select File->Remove User App from Target**
 - Select the *nasa_table* app and the option to preserve the source files



Lesson 2 – Build New cFS Target (2 of 2)



2. From Basecamp's main menu select File->Add User App from Target
 - Select the *nasa_table* app and use the green buttons to install the app



- The following slides describe how to perform table operations

Lesson 2 –Table Operations (1 of 3)



1. If not done in Lesson 1, issue a cFE *Table Dump* command to create /cf/temp.tbl

Send CFE_TBL/Application/CMD Telecommand

Command	DumpCmd	
Parameter Name	Type	Value
ActiveTableFlag	CFE_TBL/BufferSelect	ACTIVE
TableName	CFE_TBL/TableName	NASA_TABLE.ExampleTa
DumpFilename	BASE_TYPES/PathName	/cf/temp.tbl

- The following event message indicates the table was successfully written to file /cf/temp.tbl

```
EVS Port1 66/1/CFE_TBL 14: Successfully dumped Table 'NASA_TABLE.ExampleTable' to '/cf/temp.tbl'
```

Lesson 2 –Table Operations (2 of 3)



2. Issue cFE Table Load command to load /cf/temp.tbl in a working buffer

Send CFE_TBL/Application/CMD Telecommand

Command	LoadCmd	
Parameter Name	Type	Value
LoadFilename	BASE_TYPES/PathName	/cf/temp.tbl

- The following event message indicates the table was successfully loaded

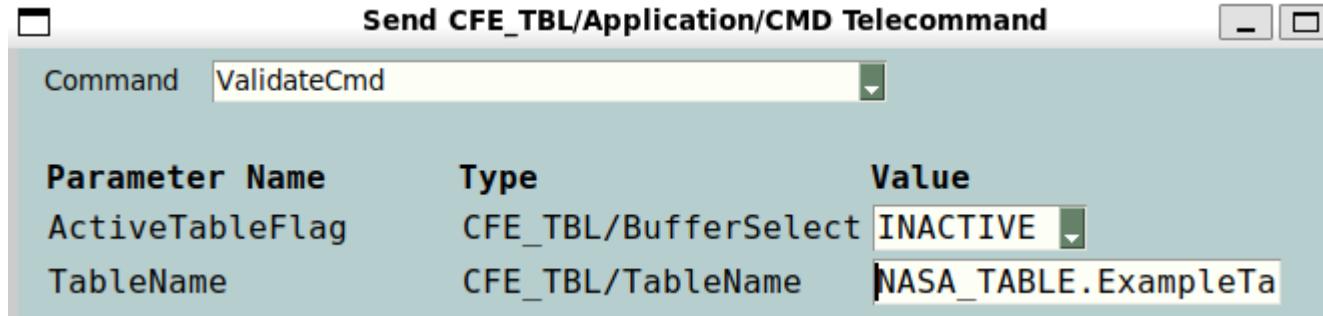
```
EVS Port1 66/1/CFE_TBL 12: Successful load of '/cf/temp.tbl' into 'NASA_TABLE.ExampleTable' working buffer
```

- Note when a table is dumped, the table name is written to the file header so the table name is not part of the load command

Lesson 2 –Table Operations (3 of 3)

3. Issue cFE Table Validate command to validate the table data while it's still in the working buffer

- The validation may take a couple of seconds because table validation is performed during the housekeeping cycle done every 4 seconds



- The following event message indicates the table was successfully validated **

EVS Port1 66/1/CFE_TBL 36: NASA_TABLE validation successful for Active 'NASA_TABLE.ExampleTable'

EVS Port1 66/1/NASA_TABLE 9: Table Int1 value 256 exceeds validation limit 20
 EVS Port1 66/1/CFE_TBL 96: NASA_TABLE validation failed for Inactive 'NASA_TABLE.ExampleTable', Status=0xFFFFFFFF

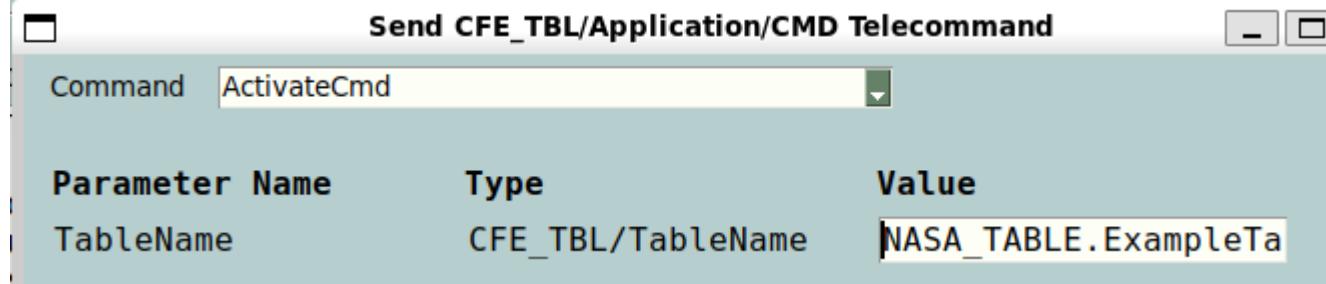
- The app's table validation function was called during this command processing

**There's a bug in Basecamp's version of the EDS Toolchain which causes the cFS binary table load & dumpsnot to work properly. See Basecamp issue #121.

Lesson 2 –Table Operations (4 of 3)



4. Issue cFE Table Activate command to cause the new table to be used



- The following event message indicates the table was successfully validated
- We did not change the table values so it should pass validation. In the next lesson, Basecamp's Table Manager tool will be used to change table values and the validation function can be tested.

Objectives

- Learn how to define a cFS Binary table in EDS
- Learn how to use Basecamp's Table Manager tool to change table values

Proxy Telemetry

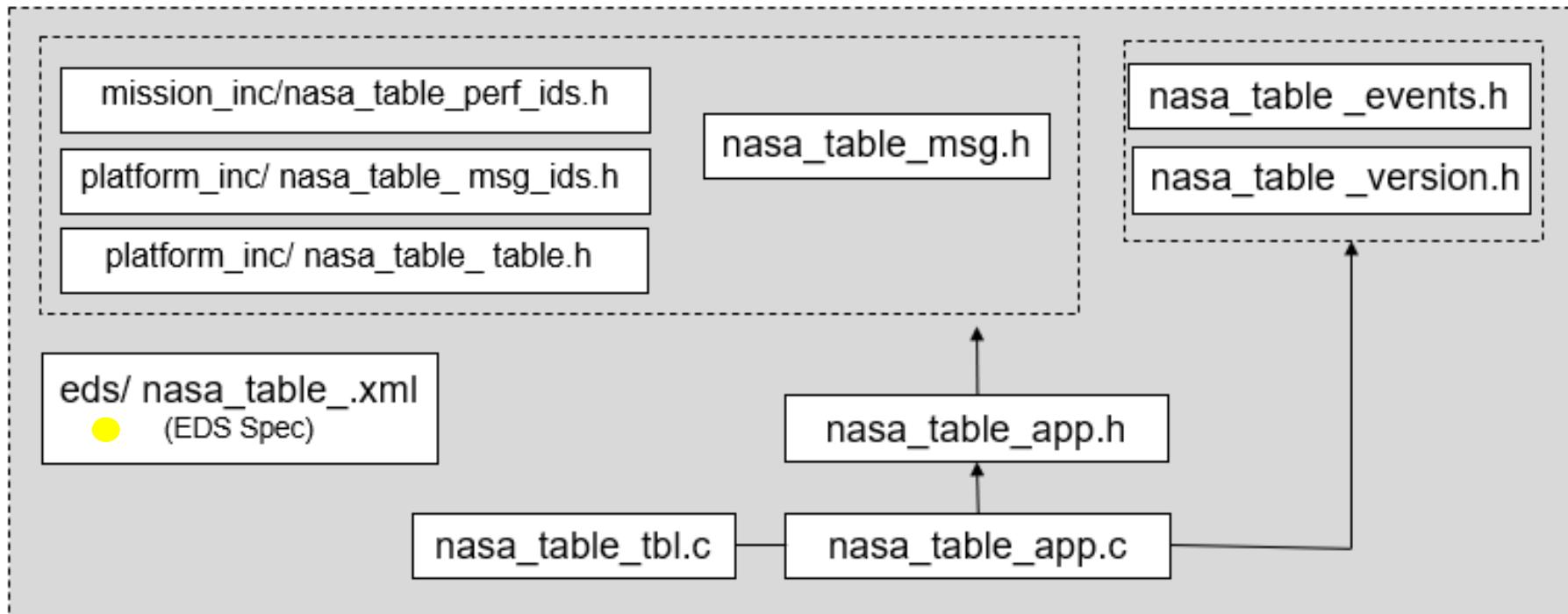
- EDS can define the binary file data content but that definition can't be exported as part of the EDS "required interface"
- The EDS version used by Basecamp only supports command and telemetry messages as part of the required interface
- Therefore the context of a *proxy telemetry* message to represent a binary table file was created
- The EDS table file limitation will be addressed in a future cFS open source release

Lesson 3 – File Modifications (1 of 3)



- The following files are modified in this lesson

App Source Files



nasa_table_tbl.xml

1. Define a telemetry message payload that is the binary table file
 - Note the current cFS provides file header definitions in CFE_FS and CFE_TBL

```
<ContainerDataType name="TblFile_Payload" shortDescription="Proxy telemetry payload">
  <EntryList>
    <Entry name="FileHdr" type="CFE_FS/Header"      shortDescription="cFE standard file header" />
    <Entry name="TblHdr"  type="CFE_TBL/File_Hdr"   shortDescription="cFE standard table header" />
    <Entry name="TblData" type="ExampleTable"        shortDescription="" />
  </EntryList>
</ContainerDataType>
```

2. Define a table file interface and by convention the name must end in 'TBL_FILE'
 - In this case 'TBL_FILE' is the complete name which is okay

```
<Interface name="TBL_FILE" shortDescription="Proxy telemetry for File System Table File" type="CFE_SB/Telemetry">
  <GenericTypeMapSet>
    <GenericTypeMap name="TelemetryDataType" type="TblFile" />
  </GenericTypeMapSet>
</Interface>
```

nasa_table_tbl.xml

3. Define a telemetry message implementation

```
<Implementation>
  <VariableSet>
    <Variable type="BASE_TYPES/uint16" readOnly="true" name="CmdTopicId" initialValue="${CFE_MISSION@TEMPLATE @_CMD_TOPICID}" />
    <Variable type="BASE_TYPES/uint16" readOnly="true" name="SendHkTopicId" initialValue="${CFE_MISSION@TEMPLATE @_SEND_HK_TOPICID}" />
    <Variable type="BASE_TYPES/uint16" readOnly="true" name="HkTlmTopicId" initialValue="${CFE_MISSION@TEMPLATE @_HK_TLM_TOPICID}" />
    <Variable type="BASE_TYPES/uint16" readOnly="true" name="TblFileTopicId" initialValue="${CFE_MISSION@TEMPLATE @_TBL_FILE_TOPICID}" />
  </VariableSet>
  <!-- Assign fixed numbers to the "TopicId" parameter of each interface -->
  <ParameterMapSet>
    <ParameterMap interface="CMD" parameter="TopicId" variableRef="CmdTopicId" />
    <ParameterMap interface="SEND_HK" parameter="TopicId" variableRef="SendHkTopicId" />
    <ParameterMap interface="HK_TLM" parameter="TopicId" variableRef="HkTlmTopicId" />
    <ParameterMap interface="TBL_FILE" parameter="TopicId" variableRef="TblFileTopicId" />
  </ParameterMapSet>
</Implementation>
```

- The Table Manager tool uses the Topic ID as a reference for the accessing the binary table definition
- The ‘TBL_FILE’ naming convention is used as a filter so table file definitions are not part of the pool of available telemetry messages

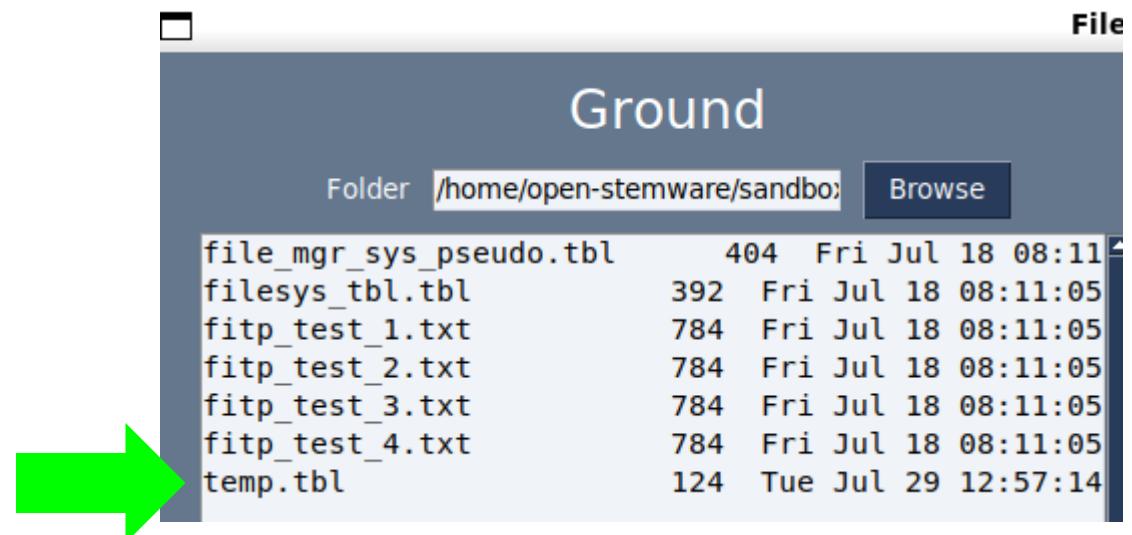
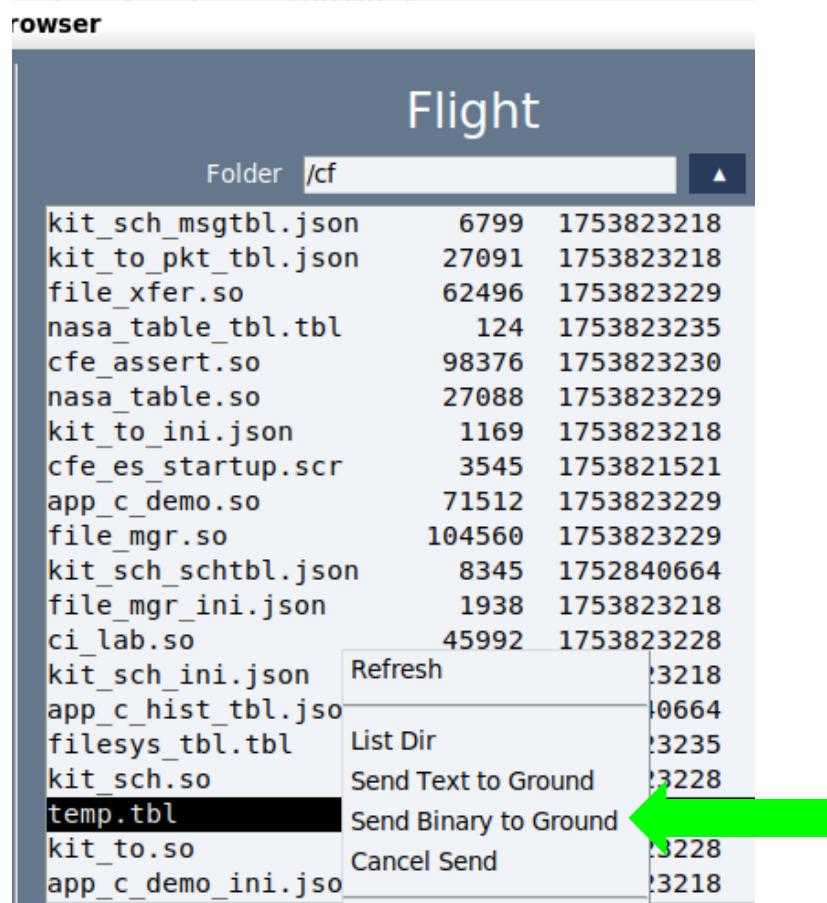
Lesson 3 – Build New cFS Target

- 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 modified, the GUI must be restarted so the new EDS library with the proxy table definition is used**
- 3. The following slides describe how to perform table operations**

Lesson 3 – Dump and Display a Table (1 of 5)

1. Use File Browser to transfer the temp.tbl file to the ground

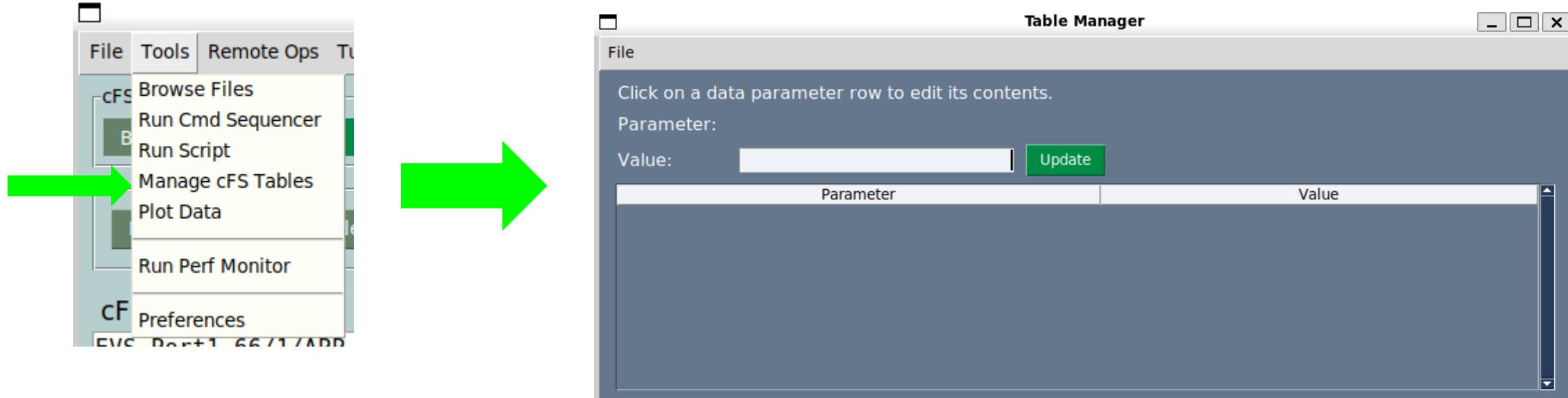
- Select *Send Binary to Ground* since table files are binary



Lesson 3 – Dump and Display a Table (2 of 5)



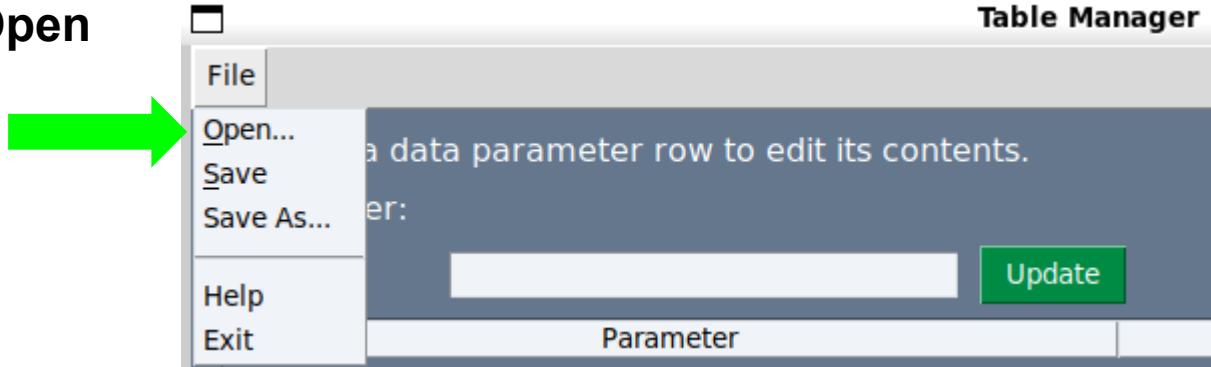
2. Launch the cFS Table Manager from the Tools menu



- This bring up an empty screen because no table file has been selected
- The next steps are to open a table file and select the proxy telemetry Topic ID that shou;ld be used to interpret the binary file

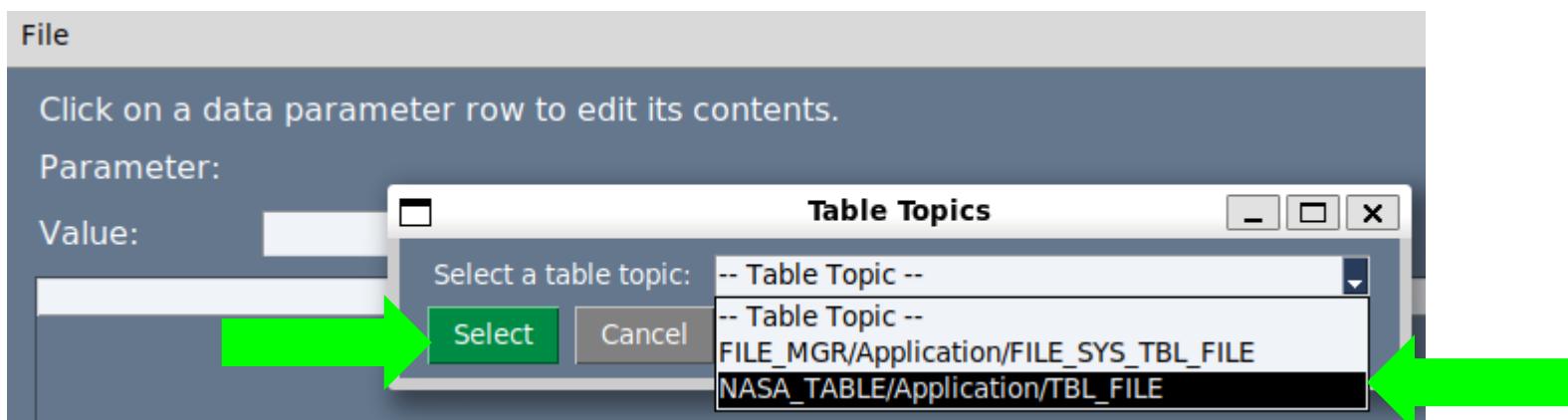
Lesson 3 – Dump and Display a Table (3 of 5)

3. Select File->Open



4. Before you select a file, you are prompted to select the Table Topic ID

- Select the NASA_TABLE TBL_FILE topic and then the <Select> button

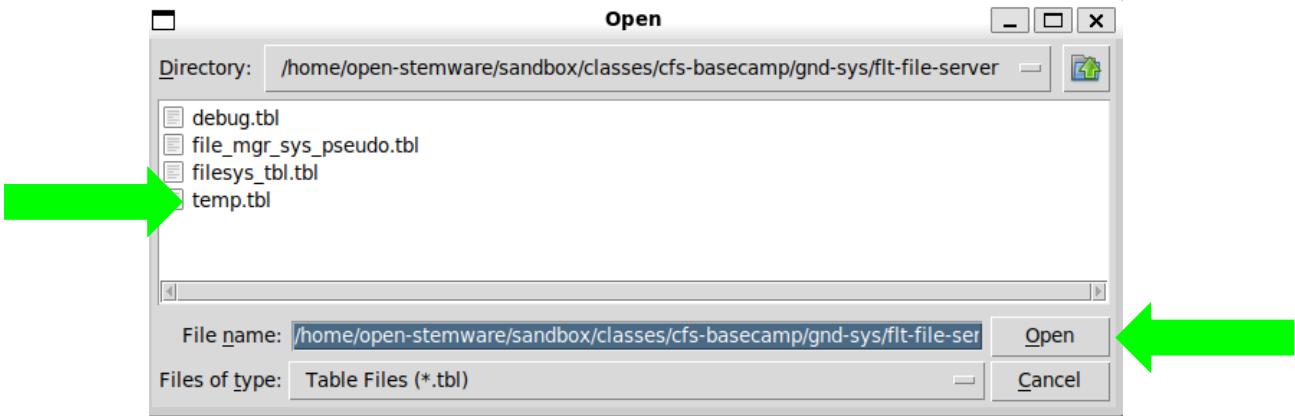


Lesson 3 – Dump and Display a Table (4 of 5)



5. After the Table Topic is selected, a file dialog box will appear

- It defaults to the flt-file-server directory which is where temp.tbl was transferred to
- Select temp.tbl and then the <Open> button



Lesson 3 – Dump and Display a Table (5 of 5)

6. The contents of temp.tbl are displayed in the <Parameter> and <Value> columns

- The TblFile.Payload.FileHgr.ContentType is the start of the table file
- The CCSDS and Sec rows are part of the proxy telemetry message
- - Scroll to the end of the file and the TblData elements will be displayed

Proxy Telemetry Header

Parameter	Value
TblFile.CCSDS.VersionId	0
TblFile.CCSDS.SecHdrFlags	Tlm
TblFile.CCSDS.AppId	106
TblFile.CCSDS.SeqFlag	3
TblFile.CCSDS.Sequence	6082
TblFile.CCSDS.Length	129
TblFile.Sec.Seconds	66051
TblFile.Sec.Subseconds	1029
TblFile.Payload.FileHgr.ContentType	1665549617

Table Data



Parameter	Value
TblFile.Payload.TblHdr.Reserved	0
TblFile.Payload.TblHdr.Offset	0
TblFile.Payload.TblHdr.NumBytes	4
TblFile.Payload.TblHdr.TableName	NASA_TABLE.ExampleTable
TblFile.Payload.TblHdr.EdsAppId	9
TblFile.Payload.TblHdr.EdsFormatId	3
TblFile.Payload.TblData.Int1	1
TblFile.Payload.TblData.Int2	2

Lesson 3 – Modify and Load a Table (1 of y)



These instructions start where the Dump and Display instructions stopped

1. Scroll down the Parameter list and select TblFile.Payload.TblData.Int entry

- The selected row is highlighted and the value is display in the Value Update field

Table Manager: temp.tbl

File

Click on a data parameter row to edit its contents.

Parameter: TblFile.Payload.TblData.Int1

Value: Update

Parameter	Value
TblFile.Payload.FileHdr.TimeSubSeconds	3231153718
TblFile.Payload.FileHdr.Description	Table Dump Image
TblFile.Payload.TblHdr.Reserved	0
TblFile.Payload.TblHdr.Offset	0
TblFile.Payload.TblHdr.NumBytes	8
TblFile.Payload.TblHdr.TableName	NASA_TABLE.ExampleTable
TblFile.Payload.TblHdr.EdsAppld	8
TblFile.Payload.TblHdr.EdsFormatId	3
TblFile.Payload.TblData.Int1	1
TblFile.Payload.TblData.Int2	2



Lesson 3 – Modify and Load a Table (2 of y)



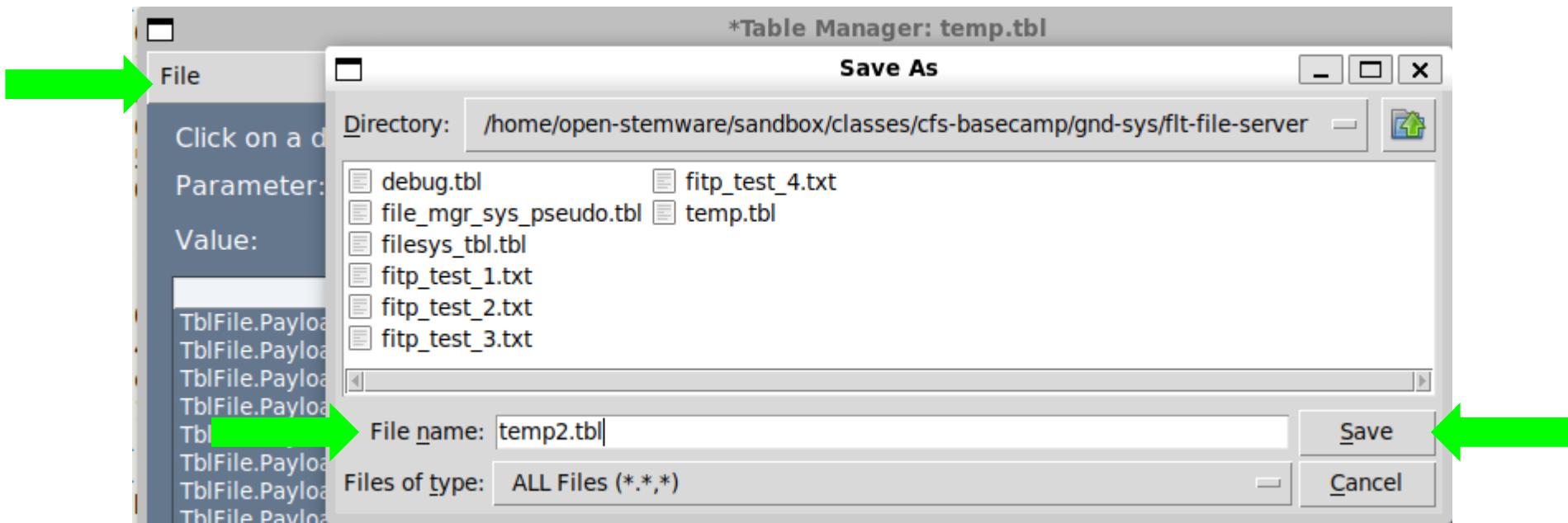
2. Change the value to 43 or any value above the maximum Int1 limit set in exercise 2 and click the <Update> button
 - The value in the table changes to 43
 - An asterisk is to the left of the file name indicating the file contents have been modified

The screenshot shows a Windows application window titled '*Table Manager: temp.tbl'. The window has a menu bar with 'File' and a toolbar with standard window controls. A large text area displays the message: 'Click on a data parameter row to edit its contents.' Below this, it shows 'Parameter: TblFile.Payload.TblData.Int1' and 'Value: 43'. To the right of the value is a green rectangular button labeled 'Update'. Below this section is a table with two columns: 'Parameter' and 'Value'. The table lists several parameters and their corresponding values. The 'Value' column for 'TblFile.Payload.TblData.Int1' is highlighted with a green arrow pointing to the right, indicating the current value being edited.

Parameter	Value
TblFile.Payload.FileHdr.TimeSubSeconds	1086681272
TblFile.Payload.FileHdr.Description	Table Dump Image
TblFile.Payload.TblHdr.Reserved	0
TblFile.Payload.TblHdr.Offset	0
TblFile.Payload.TblHdr.NumBytes	4
TblFile.Payload.TblHdr.TableName	NASA_TABLE.ExampleTable
TblFile.Payload.TblHdr.EdsApplId	9
TblFile.Payload.TblHdr.EdsFormatId	3
TblFile.Payload.TblData.Int1	43
TblFile.Payload.TblData.Int2	2

Lesson 3 – Modify and Load a Table (3 of y)

3. From the top menu select File->Save As
4. Enter a new file name such as temp2.tbl and click <Save>
 - Saving to a new filename will make it clear we are loading the modified table

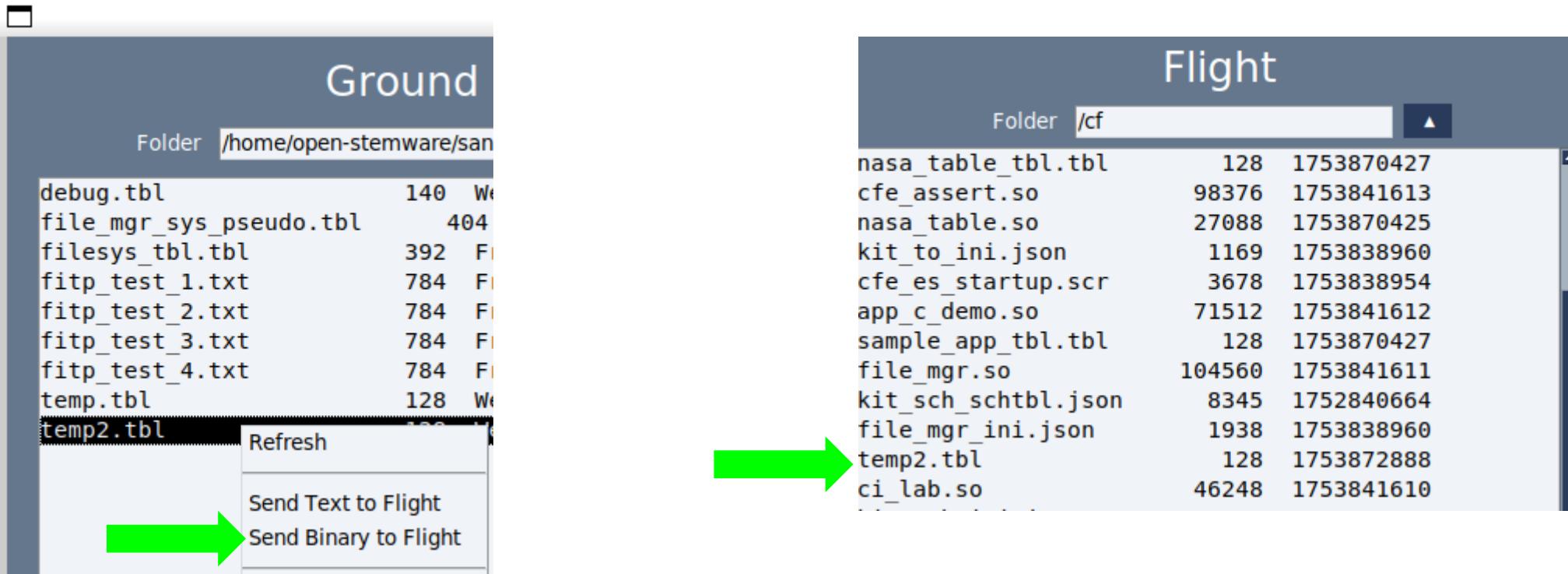


Lesson 3 – Modify and Load a Table (4 of y)

5. Close the Table Manager Tool

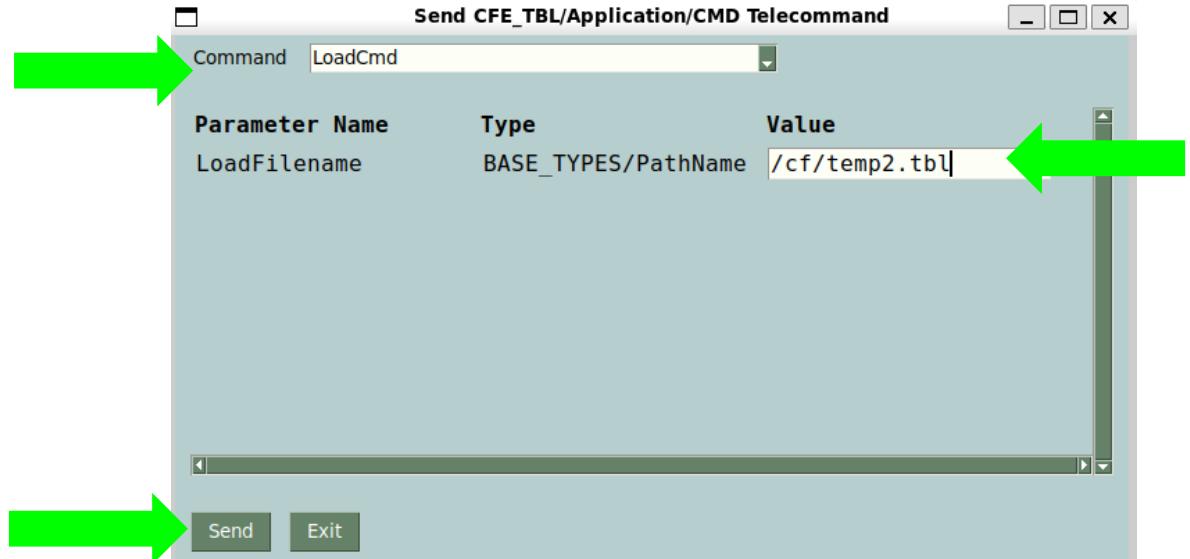
6. Use the File Browser to transfer the temp2.tbl file from ground to flight

- Select *Send Binary to Flight* since table files are binary



Lesson 3 – Modify and Load a Table (5 of y)

7. Close the File Browser
8. From the Send Cmd drop down menu select CFE_TBL->LoadCmd
9. Complete the LoadFilename entry as shown and click <Send>



- The following event message is received indicating a successful load

```
EVS Port1 66/1/CFE_TBL 12: Successful load of '/cf/temp2.tbl' into 'NASA_TABLE.ExampleTable' working buffer
```



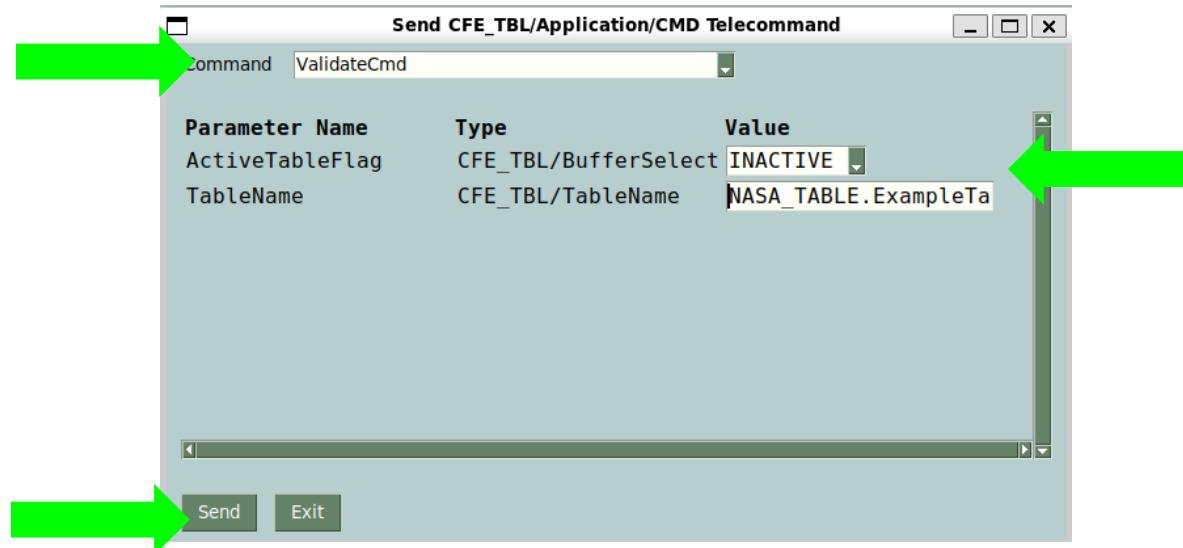
Lesson 3 – Modify and Load a Table (6 of y)



10. From the Send Cmd drop down menu select CFE_TBL->ValidateCmd

11. Complete the ValidateCmd entry as shown and click <Send>

- The Inactive buffer is the working buffer
- The table name is NASA_TABLE.ExampleTable



There is an EDS bug (see issue #121) and the validation Issue #121

```
EVS Port1 66/1/NASA_TABLE 9: Table Int1 value 16777216 exceeds validation limit 20
EVS Port1 66/1/CFE_TBL 96: NASA_TABLE validation failed for Inactive 'NASA_TABLE.ExampleTable', Status=0xFFFFFFFF
```

Lesson 3 – Next Steps

- You now have all the knowledge to work with cFS binary tables
- A good way to learn is to play around. Here are some suggestions to get you started
 - Dumping and load the same table
 - Change table values at limit boundaries to demonstrate pass/fail scenarios
 - Modify the table validation function
- The Command Sequencer tool is a helpful when you need to repeatedly send the same commands
 - It avoids the need for using the drop down menus
 - TODO: Add reference to documentation when it is available

