Table Services

Basecamp

August 2025

Audience & Prerequisites

- **Objectives**
  - Provide a comprehensive description of the cFS Framework Executive Service
- **Intended audience**
  - Software engineers developing with the cFS
- **Prerequisites**
  - cFS Framework Introduction
- **Refer to the Executive Services module for a list of topics covered in each cFE service training module**

Blue screens contain hands on cFS Basecamp exercises

- Basecamp is a lightweight environment with built-in tutorials for learning the cFS
- https://github.com/cfs-tools/cfs-basecamp

Table Services Overview

- **What is a table?**
  - Tables are logical groups of parameters that are managed as a named entity
- **Parameters typically change the behavior of a FSW algorithm**
  - Examples include controller gains, conversion factors, and filter algorithm parameters
- **Tables services provides ground commands to load a table from a file and dump a table to a file**
  - Table loads are synchronized with applications
- **Tables are binary files**
  - Ground support tools are required to create and display table contents
- **The cFE can be built without table support**
  - Note the cFE applications don't use tables

Table Service Context

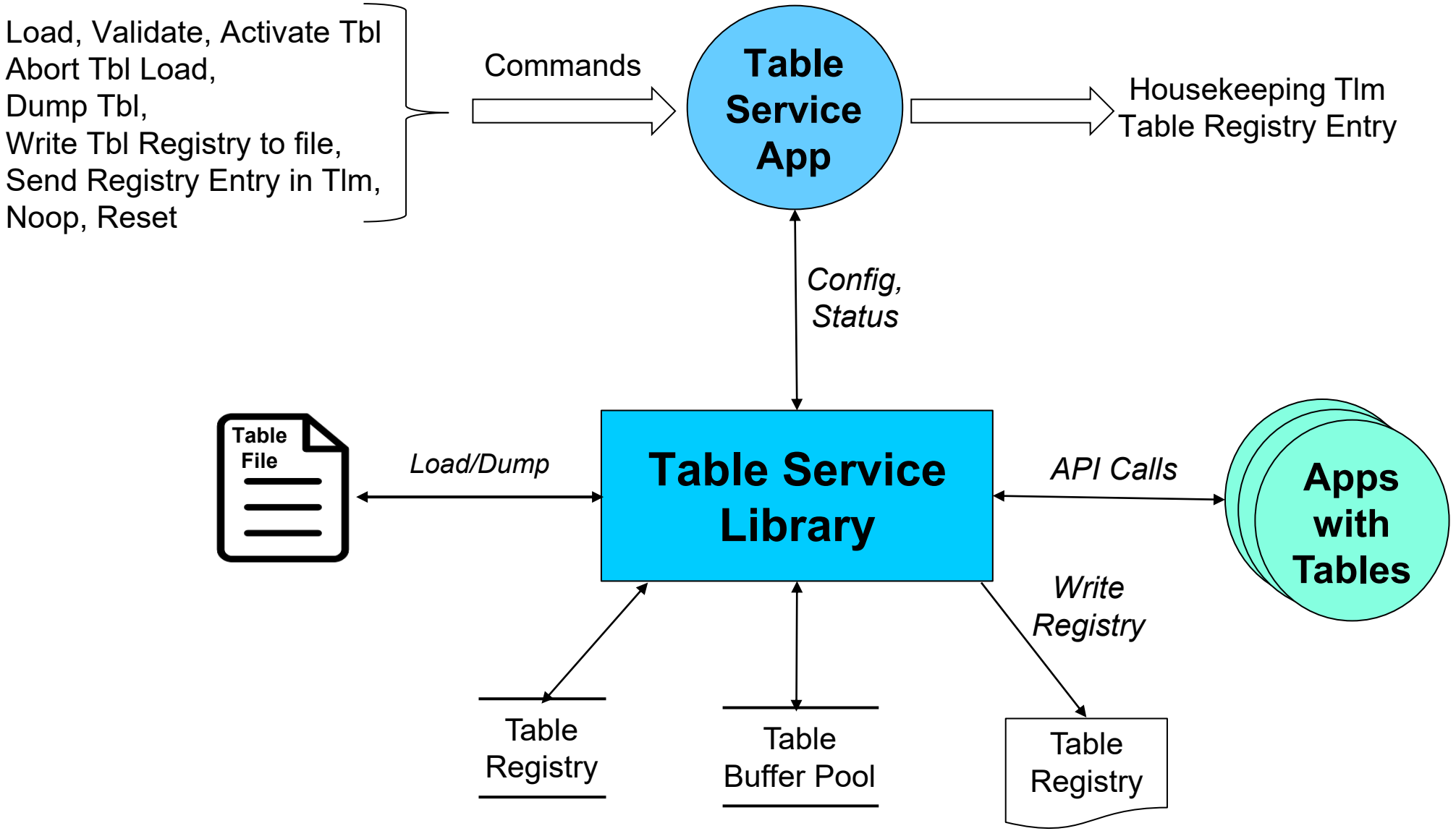
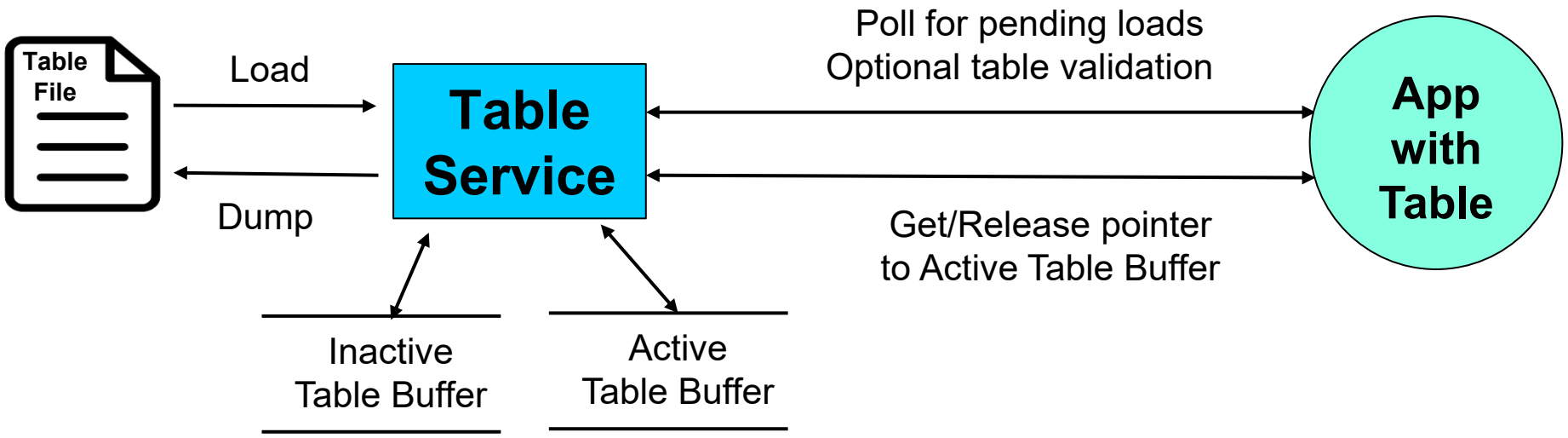
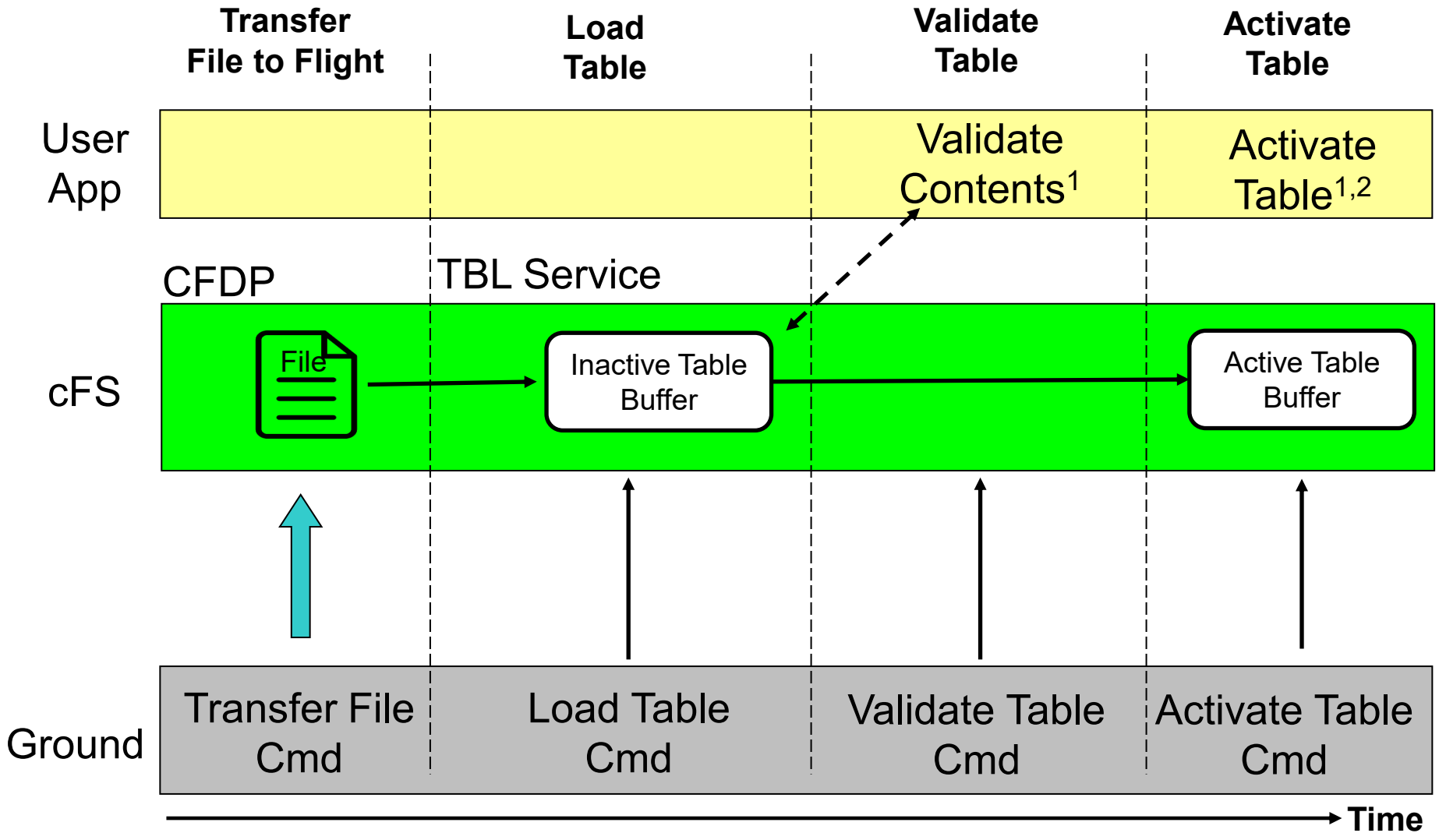


Table Service Functional Overview



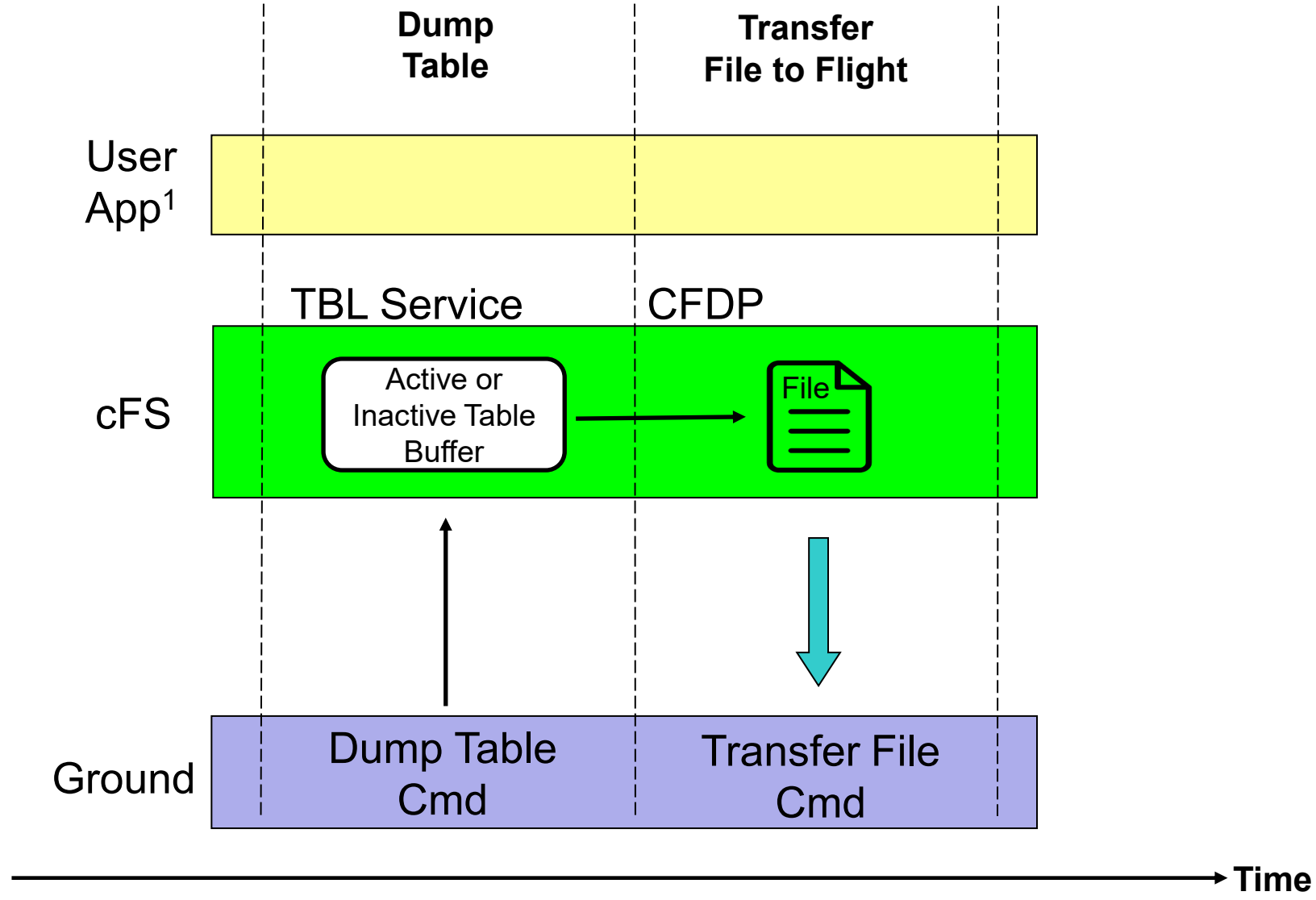
- **Table service contains buffers that hold tables for all applications**
  - Active Table Buffer - Image accessed by app while it executes
  - Inactive Table Buffer - Image manipulated by ops (could be stored commands)
- **"Table Load" is a sequence of activities to transfer data from a file to the Active Table Buffer**
- **"Table Dump" is a sequence of activities to transfer data from a either Table Buffer to a file**
- **Table operations are synchronous with the application that owns the table to ensure table data integrity**

Load Table Sequence



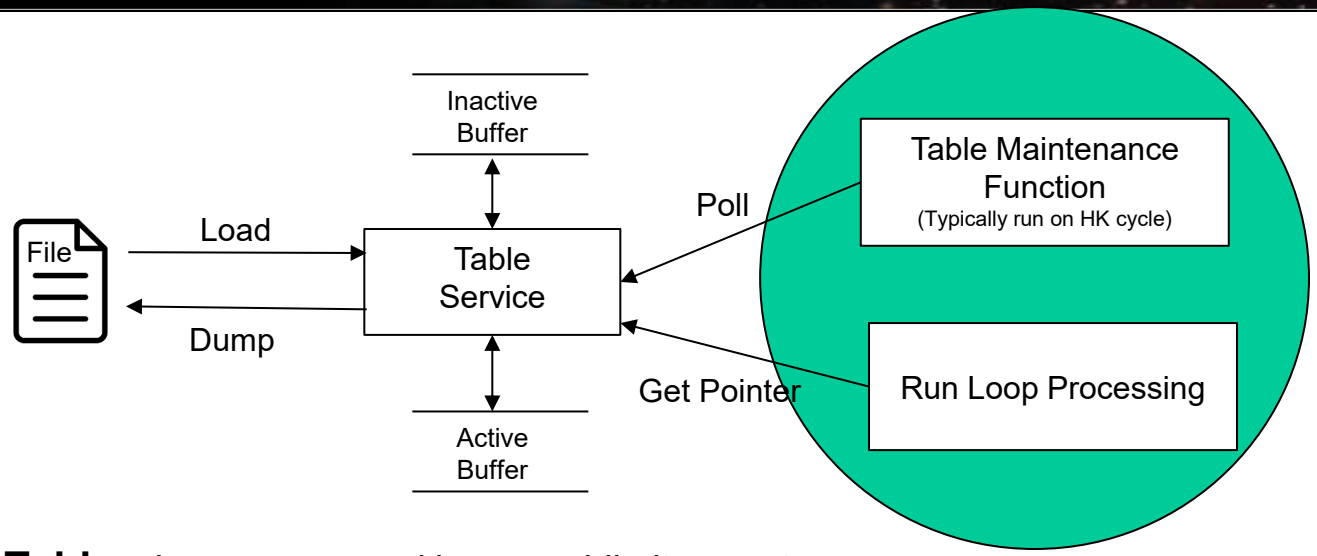
1. Apps typically validate & activate tables during their “housekeeping” execution cycle
2. In addition to instructing cFE to copy the contents, apps may have app-specific processing

Dump Table Sequence



1. Table dumps are independent of the user app

Managing Tables



- **Active Table** - Image accessed by app while it executes
- **Inactive Table** - Image manipulated by ops (could be stored commands)
- **Load → Validate → Activate**
  - Loads can be partial or complete
  - For partial loads current active contents copied to inactive buffer prior to updates from file
  - Apps can supply a “validate function” that is executed when commanded
- **Dump**
  - Command specifies whether to dump the active or inactive buffer to a file
- **Table operations are synchronous with the application that owns the table to ensure table data integrity**
- **Non-Blocking table updates allow tables to be used in Interrupt Service Routines**

Table Buffering Options: Single Buffer

- The Active Buffer is the only buffer dedicated to an application's table
- Inactive Buffers are shared to service multiple app's with single buffer tables
- Pool of fixed sized buffers that must be sized to accommodate the largest single buffer image
- *CFE_TBL_MAX_SIMULTANEOUS_LOADS* defines the number of concurrent table load sessions
- Single buffers are most efficient use of memory and adequate for most situations

Table Buffering Options: Double Buffer

- **Dedicated Inactive Buffer for each double buffered table**
- **Use when fast table image swaps are required**
- **Examples include:**
  - A high rate app
  - A very large table
  - Delayed activation of table's content (e.g. Stored Command's absolute time command table)
- **E.g. Stored Command's Absolute Time Command table**

Table Attributes (1 of 2)

- **Validation Function**
  - Applications supply a table validation function pointer when they register a table
  - The validation function is called when a Table Services *Validate* table command is processed
  - A Table Services *Activate* command will be rejected if a table has failed validation
  - Mission critical table values are usually validated

- **Critical Tables**
  - Table data is stored in a Critical Data Store so the data is preserved across Processor Resets
  - Contents updated during each table *Activate* command

Table Attributes (2 of 2)

- **User Defined Address**
  - Application provides the memory address for the active table buffer
  - Typically used in combination with a *dump-only* table

- **Dump-Only**
  - Contents can't be changed via the load/validate/activate command sequence
  - The dump is controlled by the application that owns the table so it can synchronize the dump and avoid dumps that contain partial updates

- **The *CFE_TBL_OPT_DEFAULT* macro is defined to accommodate the most common definition**

```
#define CFE_TBL_OPT_DEFAULT (CFE_TBL_OPT_SNGL_BUFFER | CFE_TBL_OPT_LOAD_DUMP)
```

Reset Behavior

- **Table registry is cleared during power-on and processor resets**
  - Applications must register tables for either type of reset
  - Applications must initialize their table data for either type of reset

- **Critical Tables**
  - If a table is registered as critical then during a processor reset table service will use Executive Services to locate and load the preserved table data from a critical data store

Retrieving Onboard State

- **Housekeeping Telemetry**
  - Table registry statistics (number of tables and pending loads)
  - Last table validation results (CRC, validation status, total validations)
  - Last updated table
  - Last file loaded
  - Last file dumped – Last table loaded

- **Telemeter Application Registry**
  - Telemeter the Table Registry contents for the command-specified table

- **Dump Table Registry**
  - Write the pertinent table registry information to the command-specified file.
  - For each table
    - Owner App ID, table name, size in bytes, attributes
    - Pointers to Active Buffer and Inactive Buffer (if double buffered)
    - Pointer to Validation function
    - Detailed table load and dump information

System Design: Guidelines

- **Tables can be shared between applications but this is rare**
  - Tables are not intended to be an inter-application communication mechanism

- **The Checksum app can be used to verify the contents of static tables have not changed**

System Design: Table Files

- **Table files are binary that contain the following three sections:**

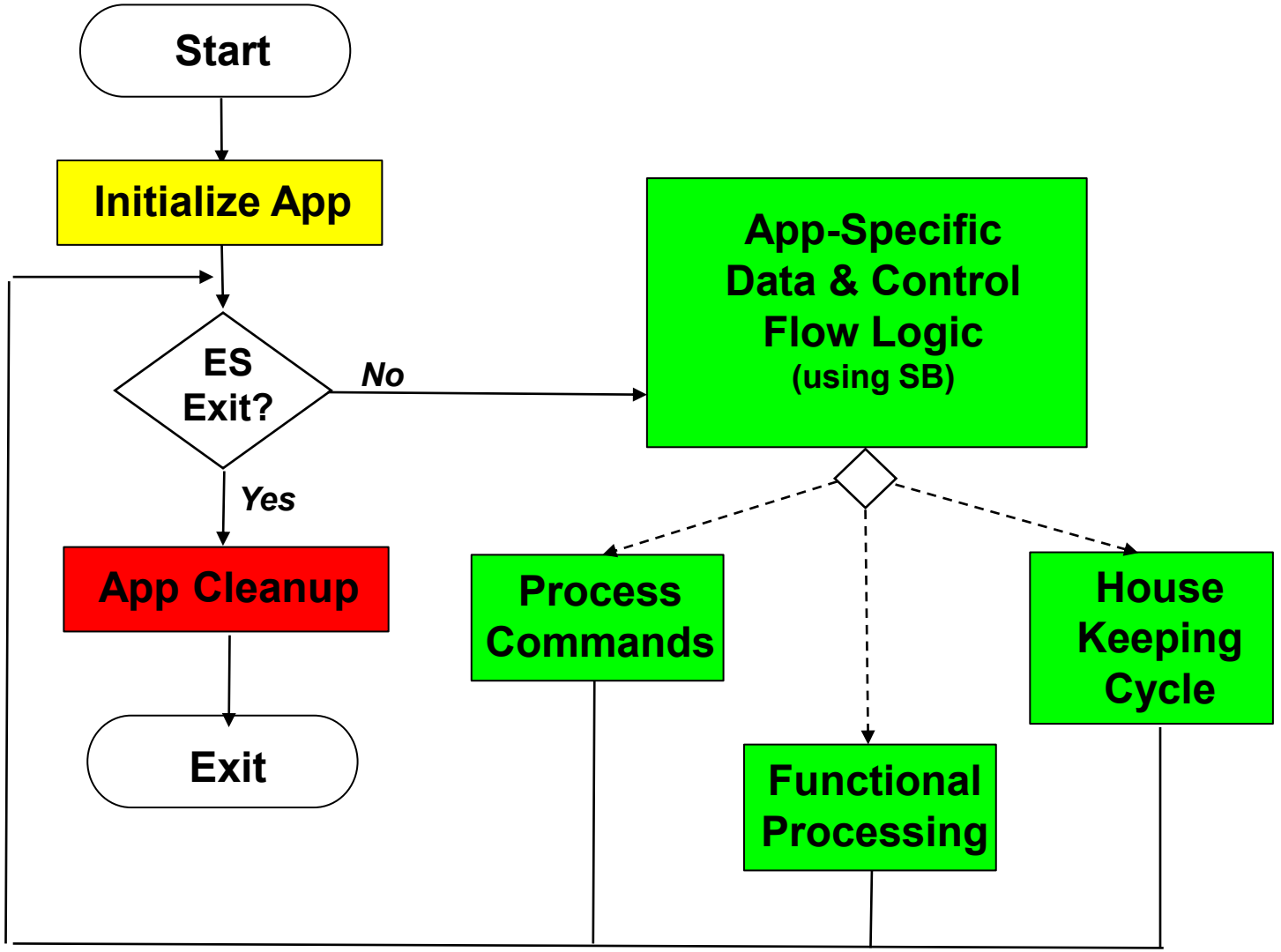
cFE File Header (cfe_fs_extern_typedefs.h : CFE_FS_Header_t)
Table Header (cfe_tbl_ extern_typedefs.h: CFE_TBL_File_Hdr_t)
Table Data (Application specific)

- **Default table files are created from C source files by the cFS cmake build system**
- **Ground tools are required to display table file contents and to modify/create new table files**
- **Table Services manages the cFE File Header and Table Header content**
  - Applications do not write to these headers

Application Development: Guidelines

- **Commands are typically used to initiate an action; not tables**
  - For example, commands are used to change the spacecraft control mode and control mode gains are defined in a table
- **Sometimes convenience commands are provided to change table elements**
  - For example, scheduler app provides an enable/disable scheduler table entry
- **Tables do not typically contain dynamic data computed by the FSW**
  - The cFE doesn't preclude this and tables have been used as a convenient method to collect data, save to a file, and transfer it to the ground
  - These are defined as dump-only tables
- **The *CFE_TBL_NotifyByMessage()* API allows an application to be notified by a software bus message when a table requires managing**
  - Avoids the need for an application to poll table services

App Development: Example Control Flow



App Development: Example Control Flow TBL Calls

Initialize App

CFE_TBL_Register() – Register app with Table Services includes an optional table validation function pointer parameter

CFE_TBL_Load() – Used to load default table values (not required)

All nominal processing flow

CFE_TBL_GetAddress() – Acquire table address prior to accessing its data

CFE_TBL_ReleaseAddress() – Release table address after finished with accessing its data

Housekeeping Cycle

CFE_TBL_Manage() – Use to coordinate table validate and activate commands

App Cleanup

CFE_TBL_Unregister() – Unregisters a table from Table Services

APIs (1 of 2)

Registration APIs	Purpose
CFE_TBL_Register	Register a table with cFE to obtain Table Management Services
CFE_TBL_Share	Obtain handle of table registered by another application
CFE_TBL_Unregister	Unregister a table

Manage Table Content APIs	Purpose
CFE_TBL_Load	Load a specified table with data from specified source
CFE_TBL_Update	Update contents of a specified table, if an update is pending
CFE_TBL_Validate	Perform steps to validate the contents of a table image
CFE_TBL_Manage	Perform standard operations to maintain a table
CFE_TBL_DumpToBuffer	Copies the contents of a Dump Only Table to a shared buffer
CFE_TBL_Modified	Notify cFE Table Services that table contents have been modified by the Application

Access Table Content APIs	Purpose
CFE_TBL_GetAddress	Obtain the current address of the contents of the specified table
CFE_TBL_ReleaseAddress	Release previously obtained pointer to the contents of the specified table
CFE_TBL_GetAddresses	Obtain the current addresses of an array of specified tables
CFE_TBL_ReleaseAddresses	Release the addresses of an array of specified tables

APIs (2 of 2)

Get Table Information APIs	Purpose
CFE_TBL_GetStatus	Obtain current status of pending actions for a table
CFE_TBL_GetInfo	Obtain characteristics/information of/about a specified table
CFE_TBL_NotifyByMessage	Instruct cFE Table Services to notify Application via message when table requires management

Command List

Command Functions	Purpose
CFE_TBL_NoopCmd	Table No-Op
CFE_TBL_ResetCountersCmd	Resets the counters within the Table Services housekeeping telemetry
CFE_TBL_LoadCmd	Loads the contents of the specified file into an inactive buffer for the table specified within the file.
CFE_TBL_DumpCmd	This command will cause the Table Services to put the contents of the specified table buffer into the command specified file.
CFE_TBL_ValidateCmd	Validate Table
CFE_TBL_ActivateCmd	Activate Table
CFE_TBL_DumpRegistryCmd	This command will cause Table Services to write some of the contents of the Table Registry to the command specified file.
CFE_TBL_SendRegistryCmd	This command will cause Table Services to telemeter the contents of the Table Registry for the command specified table.
CFE_TBL_DeleteCDSCmd	This command will delete the Critical Data Store (CDS) associated with the specified Critical Table.
CFE_TBL_AbortLoadCmd	This command will cause Table Services to discard the contents of a table buffer that was previously loaded with the data in a file as specified by a Table Load command.

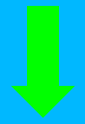
Platform Configuration Parameters

Parameter	Purpose
CFE_PLATFORM_TBL_START_TASK_PRIORITY	Defines the cFE_TBL Task priority
CFE_PLATFORM_TBL_START_TASK_STACK_SIZE	Define TBL Task Stack Size
CFE_PLATFORM_TBL_BUF_MEMORY_BYTES	Size of Table Services Table Memory Pool
CFE_PLATFORM_TBL_MAX_DBL_TABLE_SIZE	Maximum Size Allowed for a Double Buffered Table
CFE_PLATFORM_TBL_MAX_SNGL_TABLE_SIZE	Maximum Size Allowed for a Single Buffered Table
CFE_PLATFORM_TBL_MAX_NUM_TABLES	Maximum Number of Tables Allowed to be Registered
CFE_PLATFORM_TBL_MAX_CRITICAL_TABLES	Maximum Number of Critical Tables that can be Registered
CFE_PLATFORM_TBL_MAX_NUM_HANDLES	Maximum Number of Table Handles
CFE_PLATFORM_TBL_MAX_SIMULTANEOUS_LOADS	Maximum Number of Simultaneous Loads to Support
CFE_PLATFORM_TBL_MAX_NUM_VALIDATIONS	Maximum Number of Simultaneous Table Validations
CFE_PLATFORM_TBL_DEFAULT_REG_DUMP_FILE	Default Filename for a Table Registry Dump
CFE_PLATFORM_TBL_VALID_SCID_COUNT	Number of Spacecraft ID's specified for validation
CFE_PLATFORM_TBL_VALID_SCID_[1/2]	Spacecraft ID values used for table load validation
CFE_PLATFORM_TBL_VALID_PRID_COUNT	Number of Processor ID's specified for validation
CFE_PLATFORM_TBL_VALID_PRID_[1/2/3/4]	Processor ID values used for table load validation

Mission Configuration Parameters

Parameter	Purpose
CFE_MISSION_TBL_MAX_NAME_LENGTH	Maximum Table Name Length
CFE_MISSION_TBL_MAX_FULL_NAME_LEN	Maximum Length of Full Table Name in messages

Table Services Exercises - 1

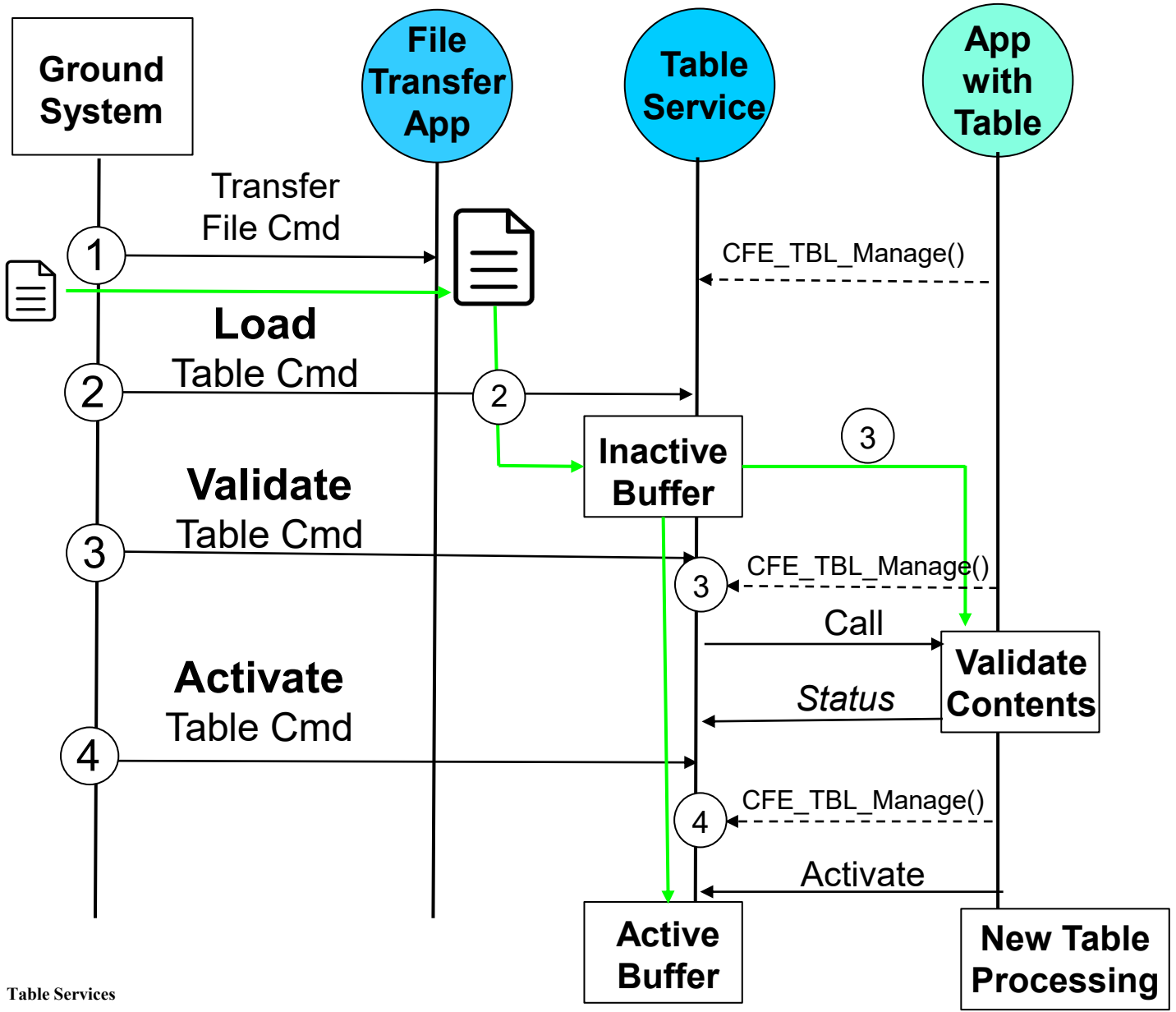


1. TODO

2. TODO

Backup Slides

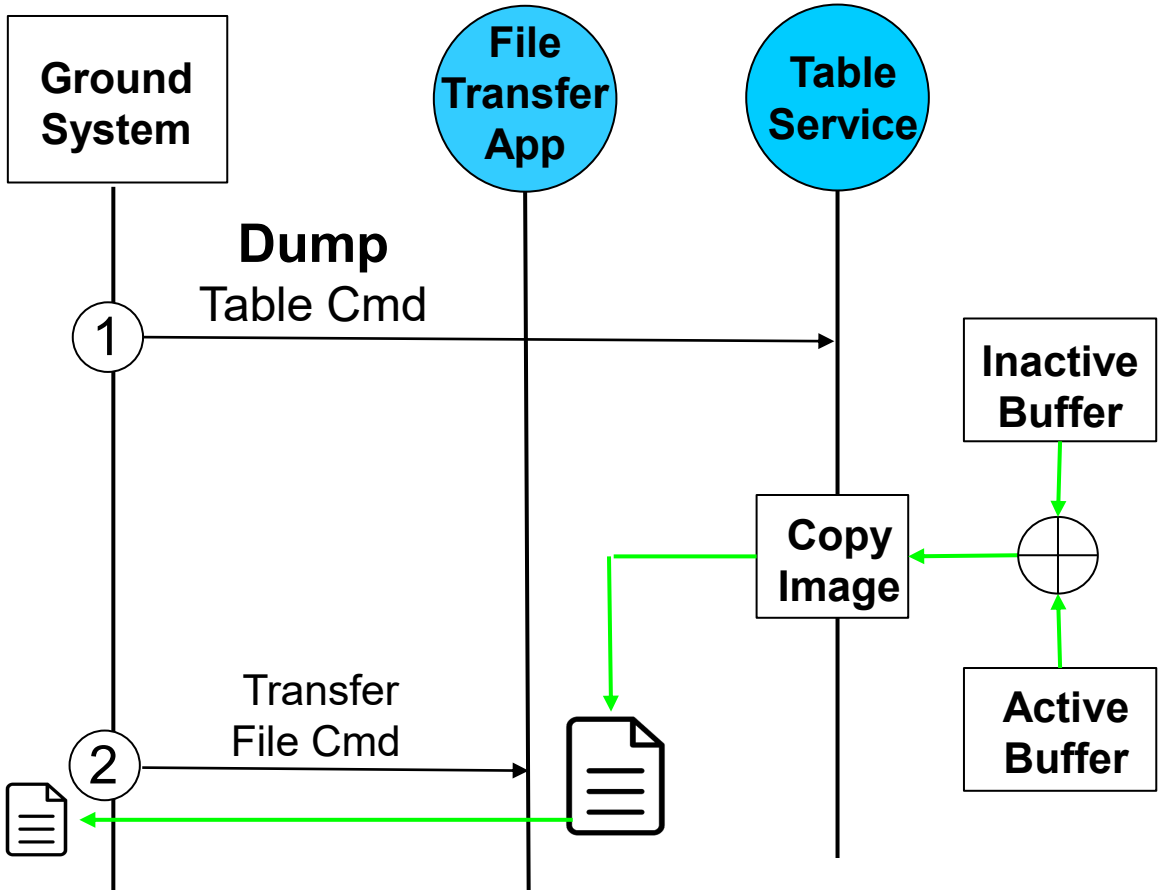
Load Table Sequence



1. **Transfer File Command**
   a. Transfer table image file from ground to flight
   b. Multiple table files can be stored onboard
2. **Load Table Command**
   a. Table Service copies file image to Inactive Buffer
   b. Loads can be partial or complete
   c. For partial loads Active Buffer contents copied to Inactive Buffer prior to updates from file (not shown)
3. **Validate Table Command***
   a. Apps validate the contents of a table image prior to table service accepting an activate command
4. **Activate Table Command***
   a. Apps initiate copy from Inactive to Active Buffer
   b. Apps may need to perform one-time functions when a new table is loaded
   c. Non-Blocking table updates allow tables to be used in Interrupt Service Routines

* Apps typically poll table services during their "housekeeping" execution cycle

Dump Table Sequence



1. **Dump Table Command**
   a. Copy either Inactive Buffer or Active Buffer to a file
2. **Transfer File**
   a. Transfer the file from flight to ground