



# Orion Backup Flight Software (BFS)

NASA/ Johnson Space Center

March 2023

Robert Hirsh robert.l.hirsh@nasa.gov

Shaurya Gupta shaurya.gupta@nasa.gov

# Agenda

- **Background**
  - Artemis and Orion
  - BFS vs. Primary Orion Software
  - cFS, CCDD, and version control
- **BFS Development**
  - Interface Management
  - Configuration Control and Workflow
  - Status and Schedule
  - Verification and Validation
- **Lessons Learned**
- **Q + A**

# Artemis Background

- ***Artemis***, the twin sister of Apollo and Goddess of the Moon and the Hunt, encompasses all of NASA's present efforts to return humans to the Moon and preparing us and propelling us on to Mars
- The Artemis program will be a series of increasingly complex missions to build a long-term human presence at the Moon for decades to come
  - Through the successful **Artemis I** mission (Nov/Dec 2022), the deep space exploration Orion spacecraft's systems were tested in a spaceflight environment, as well as ensuring a safe re-entry, descent, splashdown, and recovery prior to the first flight with crew on Artemis II.
  - **Artemis II** will be the next mission (2024) and the first flight with crew aboard of NASA's Orion spacecraft -- this will be the first crewed test flight to the Moon since the Apollo program. The Artemis II flight test will pave the way to land the first woman and first person of color on the Moon on **Artemis III** (2025).

# Orion Background

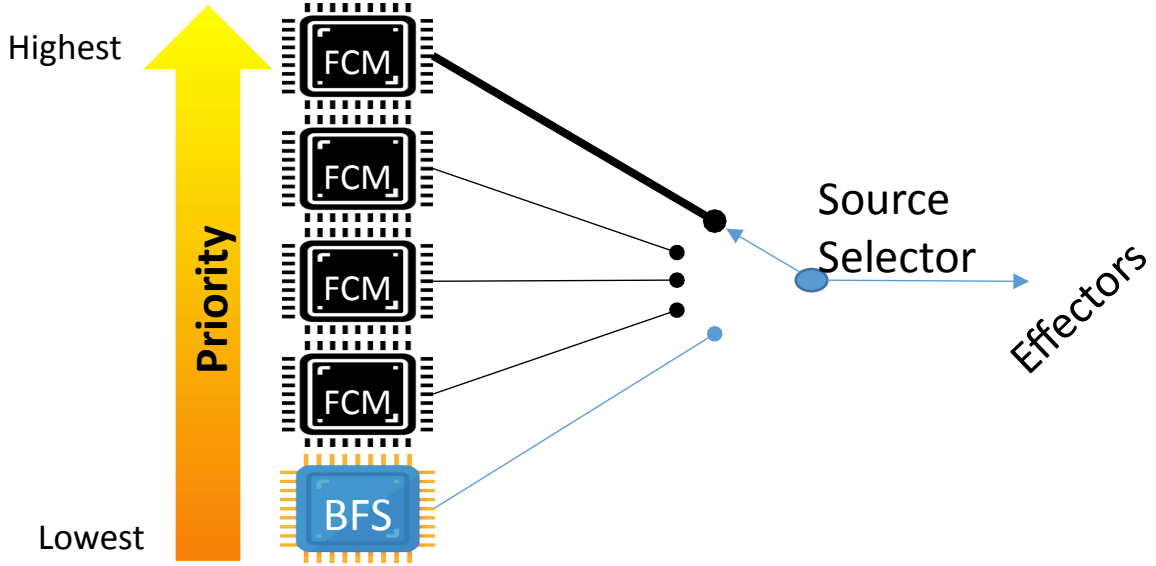
- Named after one of the largest constellations in the night sky and drawing from more than 50 years of spaceflight research and development, the Orion spacecraft is designed to meet the evolving needs of NASA's deep space exploration program for decades to come
- Orion will serve as the multi-purpose exploration vehicle that will carry crew to space, provide emergency abort capability, sustain astronauts during their missions, and provide safe re-entry from deep space return velocities
  - The **Launch Abort System** will carry the crew to safety in the event of an emergency during launch or ascent atop the agency's Space Launch System rocket
  - The **Crew Module** is the pressurized part of the Orion spacecraft where crew will live and work on their journey to the Moon and back
  - The **Service Module** provides propulsion, thermal control, electrical power generated by solar arrays, and life support systems including water, oxygen, and nitrogen

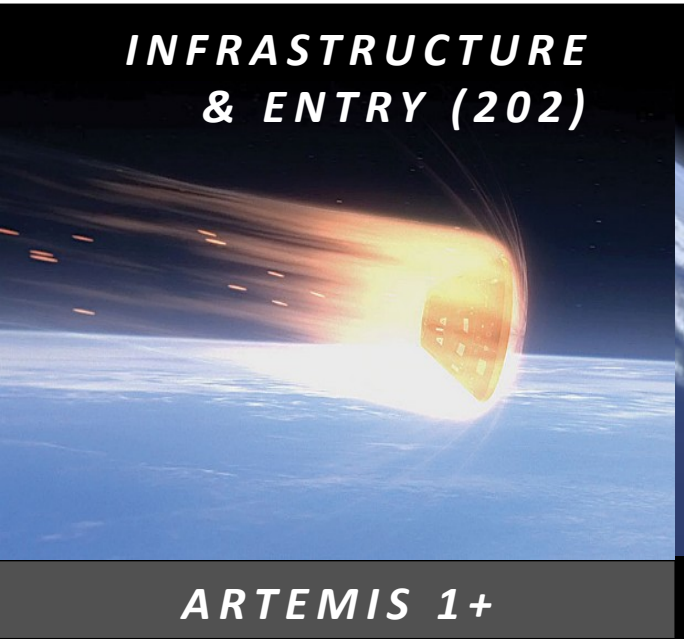
# High Level Artemis Schedule (likely to change!)

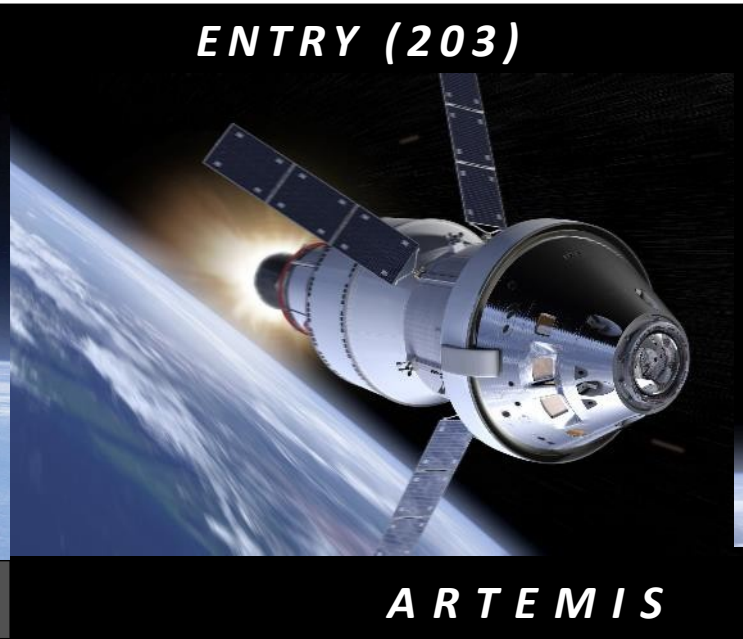
- Pad Abort I – May 6th, 2010
  - 95 second test of LAS (from rest off the launchpad)
- EFT-1 – Dec 5th, 2014 (1675 day gap)
  - 4-hour 3-orbit test of Orion Crew Module
- AA-2 – July 2nd, 2019 (1669 day gap)
  - 3-minute test of LAS (from 30,000ft going Mach 1)
- Artemis I – Nov 16th 2022 (1233 day gap)
  - ~30 day test of CM and SM (no crew)
- Artemis II (ETD: late 2024) ( ~716 day gap)
  - First Orion mission with crew, ~10 days
- Artemis III (ETD: late 2025) ( ~406 day gap)
  - First lunar landing mission since Apollo, ~30 days total

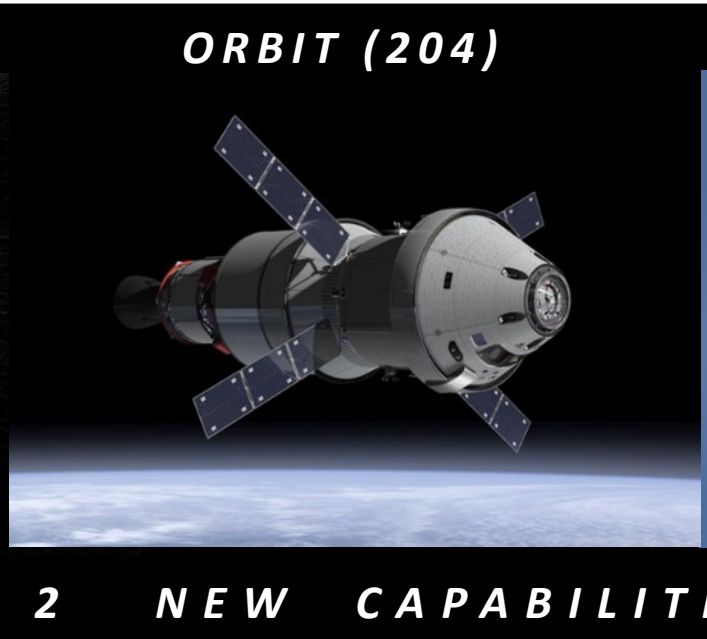
# Orion Backup Flight Software

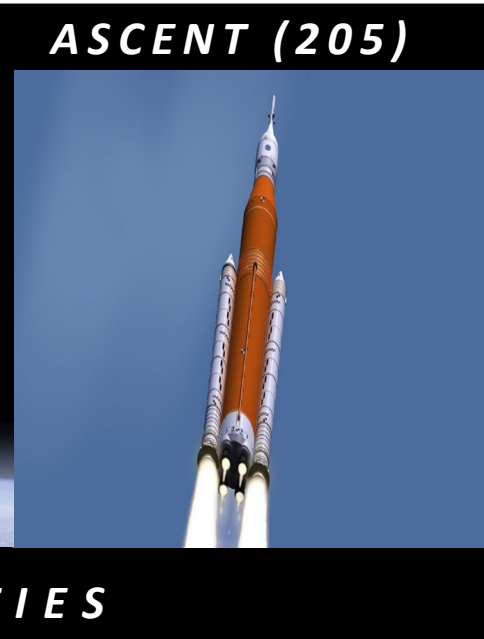
- Orion is controlled by four redundant flight computers. If all four computers fail simultaneously due to a latent software bug, then Backup Flight Software (BFS) takes control of Orion.
- BFS is a ***dissimilar*** flight software system that mitigates against a common-cause primary computer failure.
- BFS provides core capabilities to ***survive the crisis*** (3 hours) until primary FSW is restored.
- *Beresheet* (‘19), *Ariane 5* (‘96)



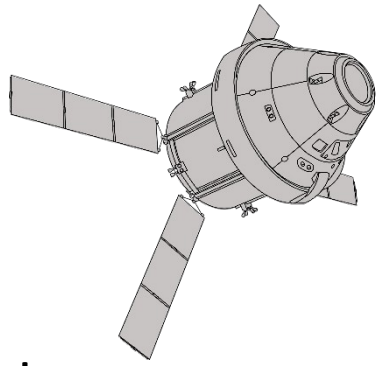




ARTEMIS 1+

ARTEMIS 2 NEW CAPABILITIES

# Backup Flight Software (BFS)

- Orion Multi-Purpose Crew Vehicle Backup Flight Software (BFS)
  - Reduces risk of “loss of crew” if Primary Flight Software (PFSW) were to fail
  - BFS is a joint NASA/LM development team
    - Different OS, HW, software, philosophy and developers
      - Like BFS used on Space Shuttle (but even more dissimilar)
    - Less complex than PFSW, and less automated (source code ~20k SLOC )
    - More responsibility on humans to send out commands when they are needed
  - BFS constantly sends commands, but BFS is ignored as long as primary is active,
    - If primary fails (or does not send commands), then BFS immediately takes over
    - “Hot backup” for dynamic phases flight (ascent, burns, and entry)
      - Times when vehicle needs to send commands sooner than PFSW can reboot
    - Lots of things must go wrong for BFS to takes over -- Hopefully BFS is never used!
    - BFS decided to use cFS architecture (unlike PFSW)




# cFS Background


- cFS (core Flight System) was developed by NASA Goddard Space Flight Center as a reusable software framework for software development (especially for spacecraft)
  - Freely available for download from https://cfs.gsfc.nasa.gov/
- cFS has been, is, and is intended to be used by many projects
  - Examples: Lunar Atmosphere and Dust Environment Explorer (LADEE), Lunar Reconnaissance Orbiter (LRO), Global Precipitation Measurement(GPM), Morpheus test bed, advanced EVA suit, Orion AA2 backup flight system, Orion BFS
- cFS does not specify a specific type of CDD (Command and Data Dictionary) for cFS projects
  - Any flight program will require some form CDD (as part of documentation if nothing else)
  - This could be done by a spreadsheet, with custom utilities to convert data into useful files
    - Custom utilities trigger project-specific development effort (little reuse potential)
    - Using a generic CDD utility can minimize this duplication of effort on each project

# CCDD Background

- cFS Command and Data Dictionary utility (CCDD) was designed as a generic utility to eliminate duplication of effort in order to make CDD management easier
  - Open source project available at https://github.com/nasa/CCDD
  - Written in Java for maximum portability (Linux/Mac/Windows)
  - Stores all BFS interface information into an open source database (PostgreSQL)
- Easy creation/modification of CDD information
  - Graphical user interface (GUI) to interact with the database
  - Limited to a single user (at a time) for a given database
  - Bidirectional transfer of information to/from the CCDD
    - Cut-n-paste to Excel, import/export via XTCE/CSV/JSON
- Easy access to CDD information (via scripting languages and web applications )
  - Allows user to write custom scripts (ruby/python/js) and access CDD information
    - Create vehicle and ground software products, data summary, etc
    - Generate complicated cFS products: Schedule or network tables, copy table, etc

# Interface Management

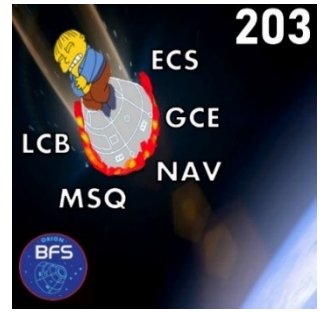
- All external interfaces for BFS are managed in CCDD
  - Autogenerates C++ header files for BFS (*.h)
  - Reports and Orion Data Services (ODS) formatted data
- Configuration Tables for BFS apps
  - All parameter values for each table parameter are stored in CCDD
  - Include validation information (date/name/status) along with flight value
  - Autogenerates of C++ table file for each BFS application
- Telemetry Packing Map (TPM) generation
  - Each TPM is defined in the CFS HK apps “copy table”
  - Frankenstein recipe to create downlink telemetry (very limited bandwidth)
- Display Packing Map (DPM) generation
  - Frankenstein recipe to create Display input
  - Each display format gets the data needed for that page
  - Quite Similar to a TPM, but sliced at the bit level (HK minimum size is 1 byte)

# Configuration Control and Workflow

1. Gitlab Issue
2. Gitlab Branch
3. Code Update
4. Automatic CI
5. Code Review (2 Approvals)
6. Merge branch into Main
7. Promotion from Git to Perforce

# BFS Progress to date

- Artemis I (Started in 2017, delivered 2020) – During Artemis I, the BFS Software was only active for entry (last few hours of mission). During entry, BFS ran and issued commands, but was not in control since all 4 Primary computers functioned nominally
- Artemis II BFS (Started Oct 2018)
  - 202(Artemis II - BFS delivered Feb 2020) - Major BFS update to Artemis II architecture. Added capability to control Solar Arrays
  - 203 (Entry - BFS delivered Oct 2020) – ECLS and Suit interfaces, Sequencing and Limit Checking, Command module RCS and parachutes control
  - 204 (Orbit - BFS delivered May 2021) – Navigation, Guidance and Control for Orbit phases, RCS and AUX engine interface for Service module
  - 205 (Launch - BFS delivered Dec 2022) – Final BFS code for Artemis II. Hazard monitoring, Launch vehicle interfaces, Upper stage rocket interface to BFS



# Orion BFS Releases

## Artemis I

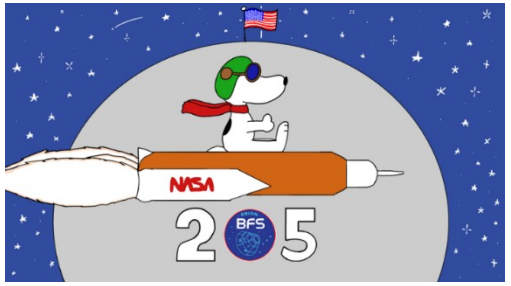
SW Dev

Oct 2020

Launched
Nov 2022

## Artemis II




SW Dev Feb 2020 Jul 2020 May 2021 Dec 2022

Launch
TBD 2024(?)

## Artemis III

304
????

SW Dev

Aug 2023

Launch
TBD 2025(?)

# BFS Schedule

- Artemis II – Launch end of 2024
  - Final BFS delivery for 205 scheduled for Aug 2023
  - Bug/Defect fixes made after 205 only as approved by Orion program
  - Simulation and training support through launch in 2024
- Artemis III – Launching in 2025
  - 303 - No new BFS content, copy existing Artemis II BFS – April 2023
  - 304 (Final BFS for Artemis III) – Adding additional Hand controller, repositioning of RCS thrusters, interface to Docking system, Jettison of docking system
    - Planning code completion in FY23, with tuning and bug fixes through the end of 2023
  - Simulation and training support through launch in 2025

# BFS Data Management Activity

- BFS (and several other Orion teams) migrated to using CCDD version 2
- Data Table management in CCDD
  - Store/document each individual parameter values (with date/signatures)
  - Sequence table management (store/document each line of each sequence)
  - Global tables (schedule tables, MIDs, etc) managed for BFS/VPU
- Data exports (each from a CCDD script)
  - Validation info (name/date/test#/etc) managed for EVERY table item
  - Information for IDD
  - Header files for all cFS SB messages used on BFS
  - Table files used by BFS (see previous bullet)
  - Telemetry management (Orion CUIs used on BFS) as well as spreadsheet to request new ones
  - Automated HK copy table (Frankenstein message for downlink)
  - Crew Display telemetry packet generation (similar to HK copy table)
- CCDD used to autogenerate/autocode a majority of all BFS SW:
  - More than 20,000 SLOC in source code headers (*.h files)
  - More than 20,000 SLOC in config tables (*.cpp files)
  - Custom “hand-coded” SLOC in BFS is less than 40,000 SLOC

# Lessons Learned

- Managing Artemis II and Artemis III at the same time is challenging
  - Most “updates” to A2 should flow into A3, but not all!
  - Perforce flowing back to gitlab caused version control conflicts
- BFS can be “forgotten about” when changes/fixes are made in Primary SW
- Being spread across 4 states made telecons a required part of BFS culture
  - This setup allowed BFS team to work from home during COVID-19 with minimal impact
- Using “frozen” cFS and VxWorks version (bleeding edge vs. end of life)
- Test as You Fly
  - Big Endian vs Little Endian
  - CRC calculation(s)
  - Test with actual HW or data recordings (documentation may not reflect reality)

# Future Work

- Artemis II config table management until launch – Sep 2024(?)
- Parallel configuration table management for Artemis III – 2025(?)
  - Different output messages (A2 vs. A3) – added telemetry and HW interface updates
  - Vehicle specific calibration values will change (mass, CG, thruster locations, etc)
- Lessons learned from using CCDD on BFS
  - Updates/Bugfixes made to CCDD for BFS ( now available to other programs)
  - Comparisons between CCDD and Rhapsody (modeling tool used on PFSW)


# Q+A