Function Specification (FncS)

IVSU_Vehicle_Function_Diff Updater

()

Document ID	583758	
Document Location	VSEM Rich Client, VSEM Thin Client	
Document Owner	Jayaraman, Vijay V. (vjayara5)	
Document Version	B	
Document Status	Released	
Date Issued	22-Oct-2019 09:30	
Date Revised	29-Oct-2019 10:48	
Document Classification	GIS1 Item Number:	
	GIS2 Classification:	

Document Approval			
Person	Role	Email Confirmation	Date

Printed Copies are Uncontrolled

Function Specification (FncS)
IVSU_Vehicle_Function_Diff Updater

Function Specification (FncS)
IVSU_Vehicle_Function_Diff Updater

Content

List of Figures

List of Tables

Function Specification (FncS)
IVSU_Vehicle_Function_Diff Updater

1 Introduction

1.1 Purpose

The purpose of the Differential Updater is to install differential updates on ECUs. The intent is for this function to interface directly with the installer on the module. This is intended to work on both modules with and without filesystems.

1.2 Scope

The following set of functions from the Global Feature & Function List is described in this specification.

Function ID	Function Name	Owner	Reference

Table 1: Functions described in this specification

1.3 Audience

The primary audience of this specification is the developer of our differential solution, and implementer of the OVTP OTA server application.

1.3.1 Stakeholder List

For the latest list of the feature stakeholder and their roles & responsibilities refer to <Put VSEM Link here>.

Role	Name
Ford in House IVI Software Technical Leader	Michael Alievsky
IVSU Product Supervisor	Brunilda Caushi
Vector OTA App Lead	Marco Wierer
VBF Spec Technical Lead	Jason Miller
Diff Updater Requirements Author	Ali Suleiman
Wind River Diff Updater	Piotr Lauk

1.4 References

1.4.1 Ford Documents

List here all Ford internal documents, which are directly related to the feature.

Reference	Title	Doc. ID	Revision
[001]	OTA_OVTP_CLIENT		
[002]	OVTP OTA Function Definition		
[004]	ECG OSInstaller Detailed Design		

1.4.2 External Documents and Publications

The list of external documents could include books, reports and online sources.

Reference	Document / Publication
[003]	DifferentialUpdaterPluginA_API_3.1.0_Draft.pdf

Table 3: External Documents and Publications

1.5 Terminology

1.5.1 Definitions

Definition	Description
Target	The location where the update will be installed, this terminology will be found in the Wind River Documentation. This is where the final result of applying a patch will ultimately be stored.
Source	The source is the prior software which the differential update will be applied to. The source software will be the active software on the module during installation.
Destination	Another word for Target, this is the language used in the Manifest Manager and within Ford. The destination where the software will be installed will be inactive during the installation.
Diff	The difference between two files. This is contains all information to build a newer version of a contiguous block of data from the source.
Differential Package	A package comprised of multiple Diffs intended to update a source with noncontiguous binary data. To newer version of software which is also non contiguous.
Contigous	Used to refer to data that occupies a single address range define by a starting and ending address. Every single address within in the range is occupied by the reference data.
Multiplicity	Multiplicity indicates what the valid number of occurences are for a field using the following the symbol: Symbol | Explanation 0 | Zero occurences 1 | One occurence ∞ | Many occurences , | Comma's indicate a logical "OR" Examples: "0,1, ∞" indicates zero, one, or many occurences. "0,1" indicates zero or one occurrences, this affectively means it is an optional field that only occurs a maximum of one time when present..
Block	A block is a contiguous binary that can be referred to with a single start address or filepath, and has a finite length.
[TBD]	Any requirements that are marked TBD still require input inorder to be implemented.
Diff Block	An operation that can be performed between a single patch and a contiguous piece of memory in Flash or a single File in a file system.
VBF Block	Data Blocks as they are define in VBF 3.1 Specification.
A/B	Indicates that an ECU has two regions of memory, and active portion and a passive portion. The active memory is where the current software that is being executed is stored. The passive memory is used for background installation. Once installation is complete, the passive and active memories switch places resulting in the newer software becoming the one present in active memory.
ABA	ABA indicates that the module has two regions of memory Active and Passive. The installation is idenetical to A/B but when activation occurs it doesn't swap, it copies from B to A instead. This means that the ecu code always executes from the same place.
AB	Used to refer to either A/B or ABA methods.

Table 4: Definitions used in this document

1.5.2 Abbreviations

Abbr.	Stands for	Description

FS	Function Requirements Specification / Function Group Specification	The document describing, collecting and developing the requirements of a function or a group of functions.

Table 5: Abbreviations used in this document.

2 Function Group Description

2.1 Overview

The Differential Updater function group consists of an Installer, a differential updater, and Block Type Handlers.
The Differential updater interfaces with the OTA Manager which is responsible for transferring the differential software package, and triggering the installation. This may be achieved by either OVTP or through the SWUM Agent. Both of these methods are leveraged by the OTA manager which is outside of the scope of this document.

2.2 Input Requirements

2.2.1 Differential SM

2.2.1.1 FUR-REQ-369552/B-###R_F_DiffUpdater_00057### signature

All requirements that use the word signature shall be interpreted as referring to the unique notation used for calling a function or function call back commonly used in C and C++.

2.2.1.2 FUR-REQ-328488/C-###R_F_Installer_SM_00007### OTA App Flash Access

Flash access shall be implemented by the OTA App and all call backs signatures shall be implemented with in the OTA App. All call back signatures shall be provided by the Diff updater supplier.

2.2.1.3 FUR-REQ-328489/B-###R_F_DiffUpdater_00038### IVI Flash Access

All required flash access call backs shall be implemented by the IVI team, and all signatures are to be provided by the supplier.

2.2.1.4 FUR-REQ-328490/C-###R_F_Installer_SM_00008### RAM limitations

RAM buffer size shall be controlled by configurable parameters set by the installer in all modules that support differential updates.

2.2.1.5 FUR-REQ-328479/C-###R_F_Installer_SM_00010### NVM Access for Pause and Resume

Installer shall provide NVM access for storing state and progress for the Differential Updater. This may be achieved, by passing a buffer to the Diff API and recording the result. The installer is always responsible for persisting the state provided by the diff updater.

2.2.1.6 FUR-REQ-328480/C-###R_F_DiffUpdater_SM_00018### Differential Updater

OTA supported ECU which has A/B or ABA method shall support differential file updates for all software that is larger than 1MB.

Memory requirements for hosting the differential updater are:

Amount of space required to store the diff update progress information (needed to resume the update): up to 512 bytes.

Required RAM available:

RAM consumption is between 5KB and 30MB depending on which compression algorithm and update strategies are used; as follow:

5KB : with LZSS compression, A/B update

40KB ~ 30MB : with LZMA compression, A/B update.

It depends on a compression level(0-12) as defined in the following table:

Compression Level	Required RAM(bytes)
0	37504
1	41600
2	49792
3	66176
4	98944
5	164480
6	295552
7	557696
8	1081984
9	2130560
10	4227712
11	8422016
12	16810624

The higher compression level will use more memory.

5KB + erased_block_size : with LZSS compression, Inplace update

40KB ~ 30MB + erased_block_size : with LZMA compression, Inplace update

It depends on a compression level(0-12).

The higher compression level will use more memory.

1.1MB ~ 18MB : with squashfs feature, A/B update only

It depends on a compression level(0-12).

The higher compression level will use more memory.

Note on SquashFS case:

UpdateSM would require more memory depending on Squash block size and Squash block count.

For example: 1.1MB is required for updating the Ford's Squash File system samples (it has 128KB - Squash File system's default value- block size and 900 blocks.)

Required ROM available:

ROM consumption for library executable is between 80KB and 200KB (also enough room for A/B partitions shall be reserved if A/B is used; this memory space depends on size of partition to be updated); as follow:

40KB : with minimum features enabled.(Release)

80KB : with all features enabled.(Release)

200KB : with all features enabled.(Debug)

2.2.2 ECU Requirements

These requirement must be implemented by the ECU

2.2.2.1 FUR-REQ-328481/B-###R_F_DiffUpdater_00044### AB Memory

The module Shall have two identical size regions of memory for any software that will support differential updates.

2.2.2.2 FUR-REQ-328482/C-###R_F_DiffUpdater_00039### Diff Update storage

The ECU shall provide a location in Flash for storing the differential package which is equal to atleast **30%** of the size of the passive B memory partition.

2.2.2.3 FUR-REQ-328483/B-###R_F_DiffUpdater_00045### Components of a differential package

A differential update shall consist of atleast one patch, and may contain a non differential portion that is written in the conventional matter. The installer shall be responsible for parsing the contents of the package. This requirement does not apply to OVTP based software updates.

2.2.2.4 FUR-REQ-328484/B-###R_F_DiffUpdater_00046### Absence of a File in a differential update for IVI module

For IVI modules In the event that a file is not present in a differential update, then the module will automatically carry over existing values for those file from active memory to the passive memory. Keep in mind this behavior is entirely different for OVTP ecu's.

2.2.2.5 FUR-REQ-328485/B-###R_F_DiffUpdater_00047### Partition Changes are not supported

When partition changes occur this shall not be supported by differential updates. The start addresses and file paths between differential updates shall always be the same. However Diffrential updates based on memory addresses shall satifiy this requirement so long as the souce and destination addresses fall in between the start address and ending address range for the AB memory.

2.2.2.6 FUR-REQ-328486/B-###R_F_DiffUpdater_00045### API Definition

The Diff Updater SM implementer shall provide the actual Application Programming interface, or actual library definition. This shall be used by Ford in House Software, Vector Team or any other Ford Supplier.

2.2.2.7 FUR-REQ-328487/C-###R_F_DiffUpdater_00058### Call Back functions implemented by the Installer

All call backs required by the Diff Updater SM shall be implemented by the supplier of the installer SM.

2.3 Assumptions & Constraints

3 Functional Architecture

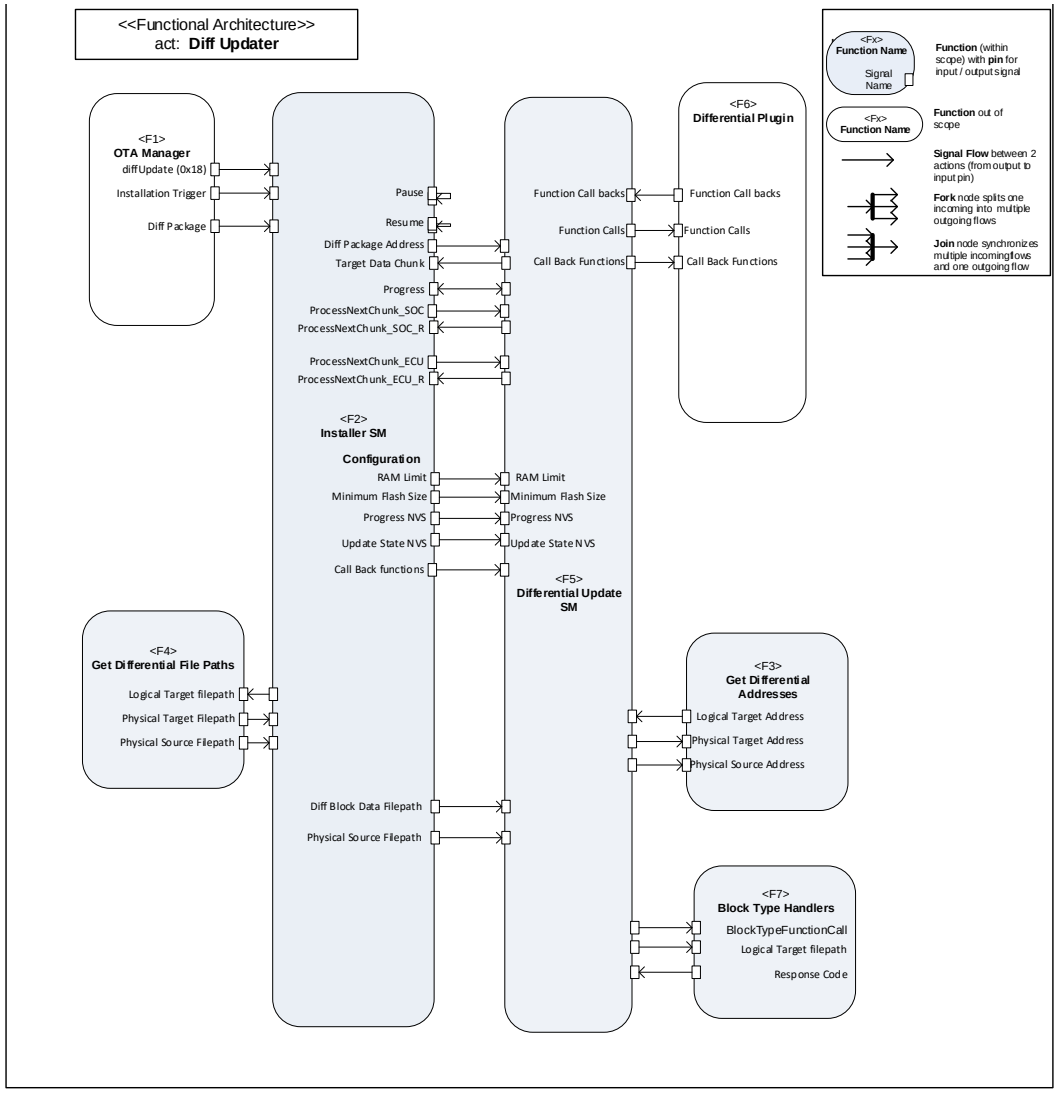


Figure 1: Diff Updater – Functional Architecture

3.1 Function List

3.1.1 List of Logical Functions

Function ID	Function Name	Function Description
F1	OTA Manager	The main controller of OTA in the vehicle
F2	Installer SM	Portion of the installer that is specifically implemented to integrate with
F3	Get Differential Addresses	Called by the Differential Update function. Returns the Physical Addresses based on a relative address.
F4	Get Differential Filepaths	Similar function to “Get Differential Addresses” but intended to be used in its place on systems with filesystems.
F5	Differential Update SM	Main state machine, new differential update triggered by FID diffUpdate (0x18) [001]
F6	Differential Plugin	The Library that is leveraged by the Differential Update SM.

Table 1: List of Logical Functions

Function ID	Function Name	FIHS	WR	Vector	Valeo
F1	OTA Manager	X			
F2	Installer SM	X		X	X
F3	Get Differential Addresses			X	X
F4	Get Differential Filepaths	X			
F5	Differential Update SM		XI		
F6	Differential Plugin		X		

FIHS – Ford In House Software
WR – Wind River
Vector –
X – Implementation
I –Interface Definition
Table 2: Function Implementation Guide

4 Logical Functions

4.1 Installer SM

4.1.1 Function Description

The Installer SM implements the portion of component that interfaces directly with the Differential Updater. This portion of the specification is intended to be used in addition to what is already defined in the OTA function definition specification, and OTA manager specification or any other document that defines installer behavior in its relation to OTA.

4.1.2 Function Interfaces

4.1.2.1 Logical Inputs

Signal ID	Signal Name	Description
00001	Installation Trigger	This is the event FID diffUpdate 0x18 , or a communication from the OTA Manager for triggering installation.
00002	Progress	Used for reporting the progress of a Differential update. Also passed and returned each time the diff updater library is called. This enables the differential update to be resumed after each individual chunk of the diff is completed.
00025	ProcessNextChunk_SOC_R	This is a response indicating the current status of the differential update. It will also include data that the installer will write to memory. This signal applies specifically to IVI modules.
00003	ProcessDiffPackage_ECU_R	This is a response indicating the current status of the differential update.
00017	Diff Package	**May contain:** LSG_DiffUpdater_00022 or LSG_DiffUpdater_00023

Formatted: Default Paragraph Font

Formatted: Default Paragraph Font

4.1.2.2 Logical Outputs

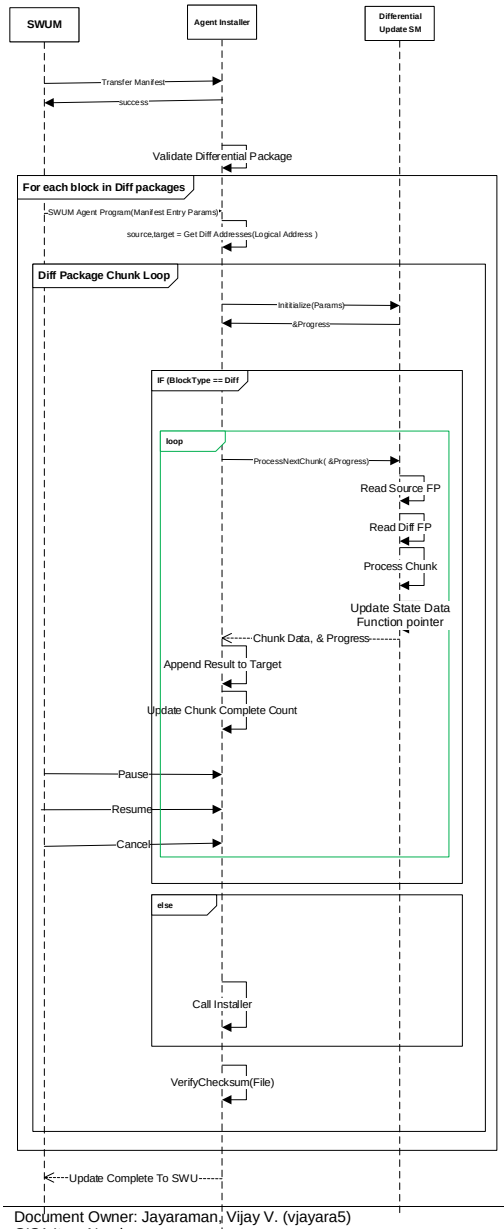
Signal ID	Signal Name	Description
	IVI	
	getRamRequired()	Call made to the Diff Updater to determine RAM required. Returns a number of bytes to allocate
00038	Inititialize(Params)	Sends the IVI diff upates params to the diff updater. This is also used for recovering state during a resume from a pause or power loss.
	ProcessNextChunk	This call is used to process the next chunk from the diff update.
	getStateData	Request for current state data so that it can be persisted. Resuming is done through the initialize param “Differential Update State Buffer” and “stateDataSize”.
	ECU	
00038	Initialize(Params)	Sends the ECU diff upates params to the diff updater. This is also used for recovering state during a resume from a pause or power loss.
	ProcessNextChunk_ECU(ChunkCount)	This call is used to process the next chunk from the diff update. **Parameters:** [in] **ChunkCount** Next chunk that is ready to process.

		Return Values: [out] **Number of Bytes** Returns Number of bytes actually read on success. Returns "0" for pending. Returns "-1" for failed response. [out] **address** Location of where result should be written. [out] **Result** (Success,Pending, Failed)
	ConfirmChunks_ECU(ConfirmationCount)	This is used to confirm that a chunk was successfully written and update the Diff updater state. This maybe used to confirm multiple chunks. **Parameters:** [in] **ChunkCount** Current ChunkCount, also the last completed chunk. **Return code** — **In progress**: More chunks still need to be written **Complete** : All chunks have been confirmed. **Error**: Chunk count out of range.

4.1.2.3 Configuration Parameters

4.1.3 Function Modeling

Figure 2: Installer SM


SWUM
Agent Installer
Differential Update SM
Transfer Manifest
success
Validate Differential Package
For each block in Diff packages
SWUM Agent Program(Manifest Entry Params*)
source,target = Get Diff Addresses(Logical Address)
Diff Package Chunk Loop
Initialize(Params)
&Progress
IF (BlockType == Diff
loop
ProcessNextChunk(&Progress)
Read Source FP
Read Diff FP
Process Chunk
Update State Data Function pointer
Chunk Data, & Progress
Append Result to Target
Update Chunk Complete Count
Pause
Resume
Cancel
else
Call Installer
VerifyChecksum(File)
Update Complete To SWU

Function Specification (FncS)
IVSU_Vehicle_Function_Diff Updater

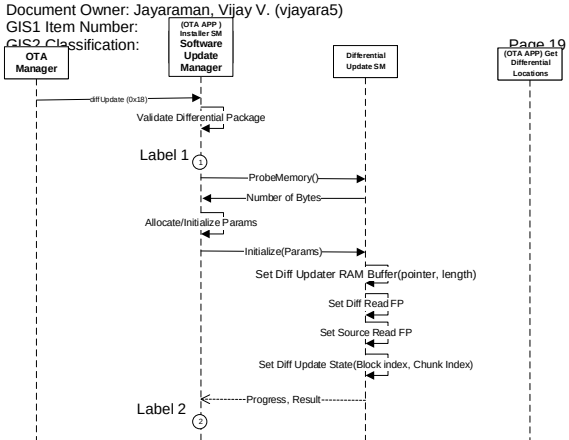


Figure 3: Installer SM for OTA App

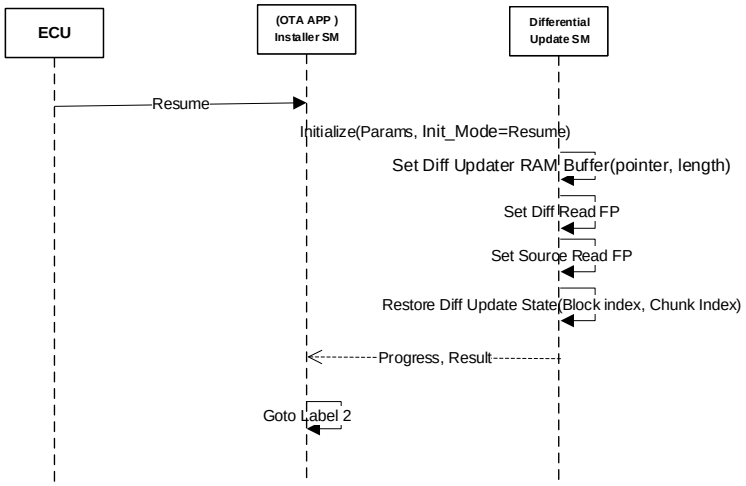


Figure 1: Resume

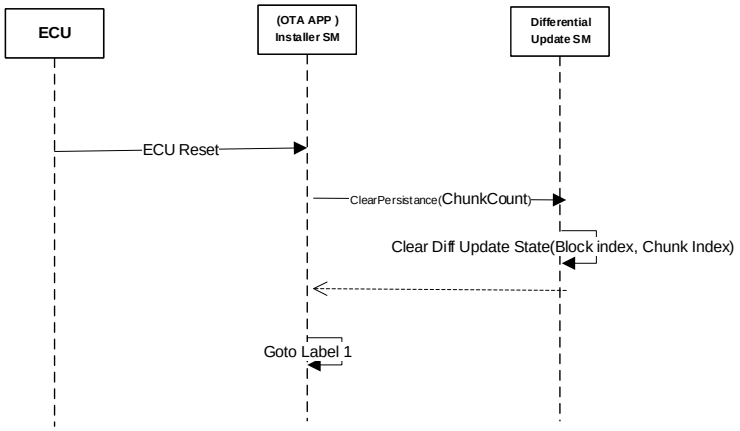


Figure 2: ECU Reset

4.1.4 Function Requirements

4.1.4.1 Functional Requirements

4.1.4.1.1 Normal Operation

4.1.4.1.1.1 FUR-REQ-328440/B-###R_F_Installer_SM_00016### Preconditions for Differential Update

The transfer and validation of the differential package shall be completed prior to the calling of the Process Next Chunk function call to the Differential Update State machine.

4.1.4.1.1.2 FUR-REQ-328441/A-###R_F_Installer_SM_00011### Memory Access to Active Memory Restriction

The Installer shall not allow the Diff Updater to write to Active Memory.

4.1.4.1.1.3 FUR-REQ-328442/A-###R_F_Installer_SM_00012### Validate Differential Block

The Installer shall validate the verificationStructureAddress of the differential package. See “diffUpdate (0x18) Function” in OVTP OTA Function Definition.

4.1.4.1.1.4 FUR-REQ-328443/A-###R_F_Installer_SM_00013### Pause and Resume

The Installer shall be responsible for Pausing and Resuming the installation.

4.1.4.1.1.5 FUR-REQ-328444/A-###R_F_Installer_SM_00017### Post Conditions

When the Diff Updater is complete, the Installer shall toggle activation in the same way that it does for non-diff updates.

4.1.4.1.1.6 FUR-REQ-328445/A-###R_F_DiffUpdater_00030### Get physical target and source for File based installers

The differential package contains a logical location for the final destination of the differential software update. When this is provided the installer shall resolve the logical locations to physical locations on the module. A location may be described as an address or file path.

4.1.4.1.1.7 FUR-REQ-328446/A-###R_F_Diff_SM_00005### Reporting Differential Update Progress

The Differential Update SM shall report differential update progress through the following metrics. Target Size, and number of bytes successfully written. Number of logical blocks written successfully written, number of blocks to be written, Number of bytes to be written in current block, total number of bytes to be written in current block.
*Status could be sent back via OVTP.

4.1.4.1.1.8 FUR-REQ-328447/B-###R_F_Installer_SM_00015### Erase Inactive Memory

The Installer SM shall erase the inactive memory prior to installing the differential update. This shall occur for each logical location where the differential updater will write to.

4.1.4.1.1.9 FUR-REQ-328448/B-###R_F_Installer_SM_00050### Common Interface Call Back Functions

The Installer shall implement all required Call backs mentioned in DifferentialUpdaterPluginA_API[005].

4.1.4.1.1.10 FUR-REQ-328449/A-###R_F_DiffUpdater_00042### No Block Reference Behavior on Modules with Filesystems

When a file doesn’t have any reference in a differential software package, the installer shall copy the active mirror of that missing file to the inactive target. This behavior results in a carry over when ever an omission occurs. This behavior only applys to differential update software packages on modules with file systems. Also take note that this shall be implemented with in the installer and not inside of the diff updater.

4.1.4.1.1.11 FUR-REQ-328450/A-###R_F_DiffUpdater_00051### Watch Dog Timers

The installer SM shall be responsible for servicing Watch Dog Timers between calls to the Diff Update SM.

4.1.4.1.1.12 FUR-REQ-328451/A-###R_F_DiffUpdater_00054### Writing result to target

The installer SM shall be responsible for writing the chunk to File or Flash Memory.

4.1.4.1.2 Error Handling

4.1.4.1.2.1 FUR-REQ-328452/A-###R_F_DiffUpdater_00047### Error Messages

All error messages in "R_F_DiffUpdater_00043" shall be translated into negative response code 0x72 "General Programming Failure" contained in [002] section 2.12. In addition to the NRC response, the module shall also update the "Last Differential Unpack Error" DID.
See "R_F_DiffUpdater_00043" for a list of error conditions

4.1.4.1.2.2 FUR-REQ-328453/A-###R_F_DiffUpdater_00050### Last Differential Unpack Error DID

When an NRC[002] error is triggered the installer shall update DID with one of the following codes based on R_F_DiffUpdater_00043. This DID shall be used to always display the last known error response. Please refer to Part II spec for details.

- $00 = No Errors
- $01 = Logical Address Required
- $02 = Data Required
- $03 = Unable to Resolve Logical Address
- $04 = Logical Source Required
- $05 = Unexpected Data
- $06 = Diff Package Invalid Format
- $FF = Diff Unpack Never Performed

4.1.4.2 Non-Functional Requirements

4.2 Differential Update SM

4.2.1 Function Description

The Differential Update State Machine provides the flow control for the Differential Updater. It is responsible for handling the differential packages and implementing all platform independent functionality.

4.2.2 Function Interfaces

4.2.2.1 Logical Inputs

Signal ID	Signal Name	Description
	IVI	
	ProbeMemory	Call made by the installer that returns number of Bytes required by the Diff Updater.
	Inititialize(Params)	Sends the IVI diff upates params to the diff updater. This is also used for recovering state during a resume from a pause or power loss.
	ProcessNextChunk	This call is used to process the next chunk from the diff update.

	getStateData	Request for current state data so that it can be persisted. Resuming is done through the initialize param “Differential Update State Buffer” and “stateDataSize”.
	ECU	
	ProbeMemory	Call made by the installer that returns number of Bytes required by the Diff Updater.
	Initialize(Params)	Sends the ECU diff upates params to the diff updater. This is also used for recovering state during a resume from a pause or power loss.
	ProcessNextChunk_ECU(ChunkCount)	This call is used to process the next chunk from the diff update. **Parameters:** [in] **ChunkCount** Next chunk that is ready to process. **Return Values:** [out] **Number of Bytes** Returns Number of bytes actually read on success. Returns “0” for pending. Returns “-1” for failed response. [out] **address** Location of where result should be written. [out] **Result** (Success,Pending, Failed)
	ConfirmChunks_ECU(ConfirmationCount)	This is used to confirm that a chunk was successfully written and update the Diff updater state. This maybe used to confirm multiple chunks. **Parameters:** [in] **ChunkCount** Current ChunkCount, also the last completed chunk. **Return code**: **In progress**: More chunks still need to be written; **Complete** : All chunks have been confirmed.; **Error**: Chunk count out of range.

4.2.2.2 Logical Outputs

Signal ID	Signal Name	Description

	IVI	
	probeMemory_response	Provides memory requirements back to the installer.
	Inititialize_Response	When the initialize API call is made to the diff updater it responds by updating the progress object, and providing a result code.
	ProcessNextChunk_Response	Response includes the Chunk Data and the progress object.
	getStateDataSize	Returns the size required for State data used by diff updater.
	ECU	
	probeMemory_response	Provides memory requirements back to the installer.
	Initialize_response	When the initialize API call is made to the diff updater it responds by updating the progress object, and providing a result code.
	ProcessNextChunk_ECU_Response	The process next chunk call responds by updating the chunk buffer, and providing an address and Result. The result may be “Pending”, “Success”, or “Failed”
	ConfirmChunks_ECU_response	The confirm chunks call response will return a result of in progress, or complete.

4.2.2.3 Configuration Parameters

Parameter ID	Parameter Name	Description
SWU Agents		
00027	Diff Updater RAM Buffer	Configurable buffer that corresponds to max flash write size. This intended for use by the Diff Updater, but is allocated by the IVI installer.
00028	Diff Updater RAM Buffer Size	Length of the Buffer, which also serves as the RAM limit for the Diff Updater. 32 bit unsigned int This value is determined by making a call to ProbeMemory on the Diff Updater by the installer.
00030	Differential Update Read Function Pointer	Location where the differential package is stored. The function will return a status of success, and failed. **Parameters:** `[in,out]` ***data** Buffer for data read from the storage. `[in]` **offset** Position from which data should be read. `[in]` **size** Length of data to be read. `[in]` ***userdata** readDiffDataFuncUserData **Return Values:** **> 0** Number of bytes actually read on success **0** 0 Read I/O is pending **-1** -1 File or Stream is ended

		-2 I/O Failure
00031	Source Read Function Pointer	Function pointer used for reading from the source on the ECU. **Parameters:** `[in,out]` **data** Buffer for data read from the storage. `[in]` **offset** Position from which data should be read. `[in]` **size** Length of data to be read. `[in]` **userdata** readSourceFuncUserData **Return Values:** **>0** Number of bytes actually read on success **0** 0 Read I/O is pending **-1** -1 File or Stream is ended **-2** I/O Failure
00032	readSourceFuncUserData	SWU Agent data for internal use.
00033	readDiffDataFuncUserData	SWU Agent data for internal use.
00034	Differential Update State Buffer	Location where the differential State is stored. This is required for resuming differential installations across power cycles. Initialzed to Null by the agent. unless the state is being resumed from a power loss.
00035	stateDataSize	Size of the state data buffer, this is defined by the Diff updater and is initialized as 0 by the Agent, unless the state is being resumed from a power loss.
ECU		
00001	Diff Updater RAM Buffer	Configurable buffer that corresponds to max flash write size. This allocated by the installer and provided to the Diff Updater.
00006	Diff Updater RAM Buffer Length	Length of buffer.
00002	Progress	The progress of the Diff update. This is read by the Installer. This includes some logging information, and the last successfully written logical address. The last logical address may be used to resolve what sector is currently being written to. This is a response that is provided by the Diff updater to the installer. Readable by the installer, written to by the diff updater.
00004	Differential Update Package Read Function Pointer	Function pointer for reading Location where the differential package is stored. This is an Asychronous call, non blocking call. The function referenced by the pointer will handle the Asychronous issues internally. The diff updater will make the same call each time until its successful, but each call will be the result of a repeated call from the installer. The Installer will handle retries. The function will return a status of pending, success, and failed. **Parameters:**

		[in,out] **buf** Buffer for data read from the storage. [in] **address** Position from which data should be read. [in] **len** Length of data to be read. [in] **Character of pointer(*) userData** readDiffFuncUserData **Return Value:** **>0** Number of bytes actually read on success **0** 0 Read I/O is pending **-1** -1 File or Stream is ended **-2** I/O Failure
00024	Source Read Function Pointer	Used by the Diff updater to read the source. This is also an Asynchronous non blocking call. The function referenced by the pointer will handle the Asychronous issues internally. The diff updater will make the same call each time until its successfull. The Installer will handle retries. The function will return a status of pending, success, and failed. **Parameters:** [in,out] **buf** Buffer for data read from the storage. [in] **address** Position from which data should be read. [in] **len** Length of data to be read. [in] **Character of pointer(*) userData** readSourceFuncUserData **Return Value:** **>0** Number of bytes actually read on success **0** 0 Read I/O is pending **-1** -1 File or Stream is ended

		-2 I/O Failure
00005	Differential Update State	Location where the differential State is stored. This is used by the Diff Updater Application, doesn't need to be parsed or read by the installer. This state should be persisted periodically the installer. Read and Writable by the Diff Updater. Number of Blocks Completed. Number of Chunks Successfully written This item is allocated by Installer. A definition for the size will be provided by the Diff updater header file.
00008	Init_Mode	Specifies whether the the Differential Update state is to be resumed or initialized.

4.2.2.4 Tunable Parameters

4.2.3 Function Requirements

4.2.3.1 Functional Requirements

4.2.3.1.1 Normal Operation

4.2.3.1.1.1 FUR-REQ-328454/A-###R_F_Diff_SM_00001### Diff Update Trigger

The Differential Update SM shall begin the differential update when it receives notification from the Installer.

4.2.3.1.1.2 FUR-REQ-328455/A-###R_F_Diff_SM_00002### Process Next Chunk Function call

When the Differential Update SM receives the Process Next Chunk function call it shall complete an atomic unit of work and return progress to the Installer.

4.2.3.1.1.3 FUR-REQ-328456/A-###R_F_Diff_SM_00003### Update complete

When the entire differential package has been returned, the Differential Update Plugin shall return a diff update complete response.

4.2.3.1.1.4 FUR-REQ-328457/A-###R_F_ Diff_SM_00019### Modules with Filesystems

For Modules that include file systems, all logical and physical file paths shall be used instead of logical and physical address.

4.2.3.1.1.5 FUR-REQ-328458/A-###R_F_ Diff_SM_00020### ECU's with out filesystems

When the Diff SM is implemented for ECUs with out File systems it shall use logical and physical addresses in place of file paths.

4.2.3.1.1.6 FUR-REQ-328459/A-###R_F_DiffUpdater_00041### Portabililty

The Differential Update SM shall be portable. This means that the code shall be compilable for all platforms as needed by Ford.

4.2.3.1.1.7 FUR-REQ-328460/A-###R_F_DiffUpdater_00033### Table for resolving logical locations to physical locations for differential updates

The Installer shall be responsible for maintaining a table that specifies the current passive and active location for a mirror on a module. This shall be updated whenever a successful activation or roll back takes place.

4.2.3.1.1.8 FUR-REQ-328461/A-###R_F_DiffUpdater_00034### Diff Block Type

When a Diff Block type is encountered by the Differential Update SM, it results in data that didn't change from the active partition block to the passive block with the addition of changes from the diff block. If the Logical Source Address is specified, then the logical source, and logical target must be resolved individually. The Logical source value is set to 0 then then only the logical target field shall be used.
The diagram below shows the active memory which is read while process a Diff block type, and the passive memory which is written to while apply the content of a diff block type.

Module A/B Memory A=A Memory Offset, B=B Memory Offset

			Active Memory (Prior to Activation) 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16			Passive Passive Memory (post Diff update) 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
DIFF	Block 0	0x01000+A	FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF	Block 0	0x01000+B	FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
		0x01010+A	FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF		0x01010+B	FF FF FF FF FF FF 00 FF FF FF FF FF FF FF FF FF
		0x01020+A	FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF		0x01020+B	FF FF FF FF FF FF FF 00 FF FF FF FF FF FF FF FF
		0x01030+A	FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF		0x01030+B	FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF

4.2.3.1.1.9 FUR-REQ-328462/A-###R_F_DiffUpdater_00035### Copy Block Type

The Copy Function shall resolve the logical address to a physical source and physical target via the table[00031][00032], then copy the contents of the physical source to the physical target. The diagram below shows the data being carried over from Active to Passive memory.

Formatted: Default Paragraph Font
Formatted: Default Paragraph Font

Module A/B Memory A=A Memory Offset, B=B Memory Offset

			Active Memory (Prior to Activation) 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16			Passive Passive Memory (post Diff update) 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
Copy	Block 1	0x01040+A	AB AB AB AB AB AB AB AB AB AB AB AB AB AB AB AB	Block 1	0x01040+B	AB AB AB AB AB AB AB AB AB AB AB AB AB AB AB AB
		0x01050+A	AB AB AB AB AB AB AB AB AB AB AB AB AB AB AB AB		0x01050+B	AB AB AB AB AB AB AB AB AB AB AB AB AB AB AB AB
		0x01060+A	AB AB AB AB AB AB AB AB AB AB AB AB AB AB AB AB		0x01060+B	AB AB AB AB AB AB AB AB AB AB AB AB AB AB AB AB
		0x01070+A	AB AB AB AB AB AB AB AB AB AB AB AB AB AB AB AB		0x01070+B	AB AB AB AB AB AB AB AB AB AB AB AB AB AB AB AB

4.2.3.1.1.10 FUR-REQ-328463/A-###R_F_DiffUpdater_00036### Write Block Type

The Write Function shall process non differential components and write them directly to the logical address in same way that a non-diff portion would be handled normally for the platform. Notice in the figure below, there is no reference to the data in the source, the result is that the all new data is written to the target.

Module A/B Memory A=A Memory Offset, B=B Memory Offset

			Active Memory (Prior to Activation) 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16			Passive Passive Memory (post Diff update) 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
Write				Block 2	0x010C0+B	E1 E1 E1 E1 E1 E1 E1 E1 E1 E1 E1 E1 E1 E1 E1 E1
					0x010D0+B	E1 E1 E1 E1 E1 E1 E1 E1 E1 E1 E1 E1 E1 E1 E1 E1
					0x010E0+B	E1 E1 E1 E1 E1 E1 E1 E1 E1 E1 E1 E1 E1 E1 E1 E1
					0x010F0+B	E1 E1 E1 E1 E1 E1 E1 E1 E1 E1 E1 E1 E1 E1 E1 E1

4.2.3.1.1.11 FUR-REQ-328464/A-###R_F_DiffUpdater_00037### Move Block Type

The Move Block type is similar to the Copy Block type except for the fact the the address of the data is different on the passive memory. The Move Block type uses the optional Source Address Field.

Module A/B Memory A=A Memory Offset, B=B Memory Offset

			Active Memory (Prior to Activation) 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16			Passive Passive Memory (post Diff update) 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
Move	Block 3	0x01100+A	CC CC CC CC CC CC CC CC CC CC CC CC CC CC CC CC			
		0x01010+A	CC CC CC CC CC CC CC CC CC CC CC CC CC CC CC CC			
		0x01020+A	CC CC CC CC CC CC CC CC CC CC CC CC CC CC CC CC			
		0x01030+A	CC CC CC CC CC CC CC CC CC CC CC CC CC CC CC CC			
				Block 3	0x01140+B	CC CC CC CC CC CC CC CC CC CC CC CC CC CC CC CC
					0x01150+B	CC CC CC CC CC CC CC CC CC CC CC CC CC CC CC CC
					0x01160+B	CC CC CC CC CC CC CC CC CC CC CC CC CC CC CC CC
					0x01170+B	CC CC CC CC CC CC CC CC CC CC CC CC CC CC CC CC

4.2.3.1.1.12 FUR-REQ-328465/A-###R_F_DiffUpdater_00044### Erase Block Type

The Erase Block type includes the start address and length fields, which are used to specify the range that needs to be clear to restart or start a diff update from the beginning. This intended to be used as a recover method the diff updater if there is a critical failure that requires a complete restart of the diff update.

4.2.3.1.1.13 FUR-REQ-328466/A-###R_F_DiffUpdater_00043### No Block Reference Behavior on Modules with out File Systems

When a Block in the source VBF doesn't have a reference in the destination VBF, this result in the block not being included in the target when the Diffrential update is applied. Any region of memory in the target that will be written shall have a reference in the diff package on modules with out File Systems.

4.2.3.1.1.14 FUR-REQ-328467/A-###R_F_DiffUpdater_00048### Copy Blocktype Function on Modules with Filesystems

On Modules with Filesystems the Copy Functionality is automaticly executed when a component of software package is not present in a differential update package. For example, if a normal Software package consists of two files A and B, and a diff created from that software package on consists of a single patch to update A, then B will be carried over from the active memory to the passive memory to be included in the update.

4.2.3.1.1.15 FUR-REQ-328469/B-###R_F_DiffUpdater_00053### Addressing for Read operations

The Differential Update SM shall provide a parameter in call back signature that indicates whether the read operation is intended for Source or Target when calling the platform implemented "Diff Read Data" Call Back s function pointer. This function pointer signature shall be provide by the diff updater supplier.

4.2.3.1.2 Error Handling

4.2.3.1.2.1 FUR-REQ-328470/A-###R_F_DiffUpdater_00043### Error Conditions

All Diff Block types shall report the first error code with the highest value that is qualified by the table below:

Error Code	Block Type	Logical Source Address Present?	Logical Target Address Present?	Data Present?	Record Present in AB Table*?	Success
0x00						No Errors
0x01	DCMWE	X	N	X	X	Logical Address required
0x02	DW	X	X	N	X	Data Required
0x03	DCM	X	X	X	N	Unable to Resolve Logical Address.
0x04	M	N	X	X	X	Logical Source Required.
0x05	CME	X	X	Y	X	Unexpected data.
0x06	DCMWE					Diff Package Parse Error, invalid Format.
0xFF						

Key:
D – Diff Block Type
C – Copy Block Type
M – Move Block Type
W – Write Block Type
E – Erase Block Type
X – Don't care
N – No
Y – Yes
*"Record in Data Table?" is an indication of whether the logical address is actually AB capable.

4.2.3.2 Non-Functional Requirements

4.3 Get Differential Addresses

4.3.1 Function Description

This function is used by the installer to resolve logical addresses. This implemented by the OTA App. This function is not directly available as a call back to the Differential Update SM, it is included to ensure that the OTA App is responsible for resolving logical addresses to physical addresses.

4.3.2 Function Interfaces

4.3.2.1 Logical Inputs

Signal ID	Signal Name	Description
00012	Logical Target Address	This is a logical address that remains the same regardless of which mirror is in use.

4.3.2.2 Logical Outputs

Signal ID	Signal Name	Description
00013	Physical Target Address	The physical target address based on which mirror is currently in use. This is where the diff updater will write the new software.
00014	Target Size	This is the maximum size that may be written to the target.
00015	Physical Source Address	The physical source address corresponds to the old software on the module. This will be read by the Diff Differential Updater, and used in conjunction with the Diff to produce the new software.
00016	Source Size	This is the size of the source that the diff is being applied to.

4.3.3 Function Requirements

4.3.3.1 Functional Requirements

4.3.3.1.1 Normal Operation

4.3.3.1.1.1 FUR-REQ-328471/C-###R_F_Installer_SM_00024### Get Differential Addresses Successful Response

The Get Differential Update Addresses function shall return the physical target and source address for a block when provided a valid logical address for a block.

4.3.3.1.1.2 FUR-REQ-328472/C-###R_F_Installer_SM_00025### Modules with out File Systems

The Get Differential Addresses function shall be implemented and provided along with all installers for modules with out file systems.

4.3.3.1.1.3 FUR-REQ-328473/B-###R_F_Installer_SM_00032### Maintaining a logical Storage for Resolving Logical Addresses

The Installer and the Get Differential addresses function shall have the ability to resolve logical addresses to physical address, for example a solution could be to use a look up table:

Logical Path	End Address	Active (Physical Source)	Passive (Logical Source)
0x20000	21FF7	0x20000 + C0000	0x20000
0x22000	221FB	0x22000 + C0000	0x22000
0x24000	B0000	0x24000 + C0000	0x24000

4.3.3.1.2 Error Handling

4.3.3.1.2.1 FUR-REQ-328474/B-###R_F_Installer_SM_00048### Unrecognized Logical Address

If the address is not resolvable, then the call back shall return and error code indicating the logical address is not valid.

4.3.3.2 Non-Functional Requirements

4.4 Get Differential Filepaths

4.4.1 Function Description

This function is used by the installer to determine the filepaths needed for applying a differential software update on a module that has a file system. It take one logical input and returns a Physical Target File path, and a Physical Source File path. This behavior is entirely implemented by the platform so that it understands the relationship between logical address and physical addresses.

4.4.2 Function Interfaces

4.4.2.1 Logical Inputs

Signal ID	Signal Name	Description
00018	Logical Target filepath	This is a logical filepath that remains the same regardless of which mirror is in use. This file path will be the same as the active physical path on the system.

4.4.2.2 Logical Outputs

Signal ID	Signal Name	Description
00019	Physical Target Filepath	The physical target filepath to the mirror not currently in use. This is where the diff updater will write the new software.
00014	Target Size	This is the maximum size that may be written to the target.
00020	Physical Source Filepath	The physical source filepath is the location of the software on the module that is currently running. This will be read by the Diff Differential Updater, and used in conjunction with the Diff to produce the new software.
00016	Source Size	This is the size of the source that the diff is being applied to.

4.4.3 Function Requirements

4.4.3.1 Functional Requirements

4.4.3.1.1 Normal Operation

4.4.3.1.1.1 FUR-REQ-328475/B-###R_F_Installer_SM_00028### Get Differential Filepaths Successful Response

The Get Differential Filepaths function shall return the physical target and source file path for a block when provided a valid logical file path.
For example, the module receives a differential software update with a logical target “/fs/voice”, the function responds with the physical target “/fs/voice_new”, and the physical source “/fs/voice”. This is because the intended destination for the update after activation is “/fs/voice” but it is still in use by the old software. Once the diff is successful, the software will toggle from “/fs/voice_new” to “/fs/voice”.

4.4.3.1.1.2 FUR-REQ-328476/B-###R_F_Installer_SM_00029### Modules with Filesystems

The get Differential Filepaths function shall be implemented for modules with File Systems.

4.4.3.1.1.3 FUR-REQ-328477/B-###R_F_Installer_SM_00031### Maintaining a logical Storage for Resolving Filepaths

The Installer and the Get Differential Filepaths function shall share a logical table that maps each logical target filepath to a physical source filepath and physical target filepath. Please see the example below:

Logical Path	Active (Physical Source)	Passive (Logical Source)
/fs/OS_System	/fs/BlockDevice1	/fs/BlockDevice2
/fs/apps/Nav	/fs/apps/Nav	/fs/apps/Nav_new
/fs/apps/Voice	/fs/apps/Voice	/fs/apps/Voice_new

4.4.3.1.2 Error Handling

4.4.3.1.2.1 FUR-REQ-328478/A-###R_F_DiffUpdater_00049### Invalid Logical Path

If the physical file path cannot be resolved the Get Differential File Paths function shall return and error code to the caller.

4.4.3.2 Non-Functional Requirements

5 Open Concerns

ID	Concern Description	e-Tracker / Reference	Responsible	Status	Solution
1	Validate Logical Block, should this be done for Diff memory, inactive, or both? and should these be done in separate steps or together. Recommendation: Validate logical block for Diff part of DiffUpdate FID Send DiffUpdate FID Validate logical blocks for Passcive				Current OVTP function definition requires the FID 18 to validate the diff, and an additional call to validate the result of the diff being applied. There is currently no is requirement to validate the source.
2	Need to add passthrough/Add Blocks scenerio				Done.
3	Need to add Block remove scenario				Done.
4	Need to follow up with Vector with new requirements update.				Done, Vector still wants a predominantly call back based interface.
5	Reduce/separate some of the interfaces for the installer in the spec, not all of these items apply to Vector.				Complete, reviewed there is only two incoming signals that are not applicable to both SOC and ECU. Added implementation guide.
6	Revisit resolving logical addresses, this could be done by adding a definitions for write operations and including a param for Active/Passive. The assumption is that the Diff Generator has no knowledge of current Active vs Passive addresses on the target ECU.				Done, added ###R_F_DiffUpdater_00053, based on current api this is the only call back that may be used in Source or Target.
7	to update Requirements for Block type functions to ensure they are controlled via Chunk size field in the diff packages.				Added ###R_F_DiffUpdater_00052
8	Servicing Watch Dog Timer				DiffUpdater_00051: The installer SM shall be responsible for servicing Watch Dog Timers between calls to the Diff Update SM.
9					Removed Duplicate error info from DiffUpdater_00034
10					UPDATED ###R_F_DiffUpdater_00043 CHANGED 0x03 to unable to Resolve Logical Address
11	Need time estimator for applying DIFF on ECU				
12	What is E2E testable?				###R_F_Diff_SM_00001### Diff Update Trigger ###R_F_Diff_SM_00003### Update complete ###R_F_DiffUpdater_00034### Diff Block Type ###R_F_DiffUpdater_00035### Copy Block Type ###R_F_DiffUpdater_00036### Write Block Type ###R_F_DiffUpdater_00037### Move Block Type ###R_F_DiffUpdater_00044### Erase Block Type ###R_F_DiffUpdater_00043### No Block Reference Behavior on Modules with out File Systems ###R_F_DiffUpdater_00048### Copy Blocktype Function on Modules with Filesystems

ID	Concern Description	e-Tracker / Reference	Responsible	Status	Solution
					###R_F_DiffUpdater_00043### Error Conditions See figure for additional details: Figure 2: Installer SM Figure 3: Installer SM for OTA App Figure 5: ECU Reset

6 Revision History

Rev. (revision)	Vers.	Date	Description	Approved by	Responsible
001			*Initial version*		
1.0.14			*Merged all local version changes into VSEM*		

7 Appendix

7.1 Data Dictionary

7.1.1 Logical Signals

7.1.1.1 FUR-REQ-328491/A-###LSG_DiffUpdater_00001### Installation Trigger

This is the event FID diffUpdate 0x18 , or a communication from the OTA Manager for triggering installation. When updating an IVI module this will be Installation trigger via SWA from the OTA Manager.

7.1.1.2 FUR-REQ-328492/B-###LSG_DiffUpdater_00002### Progress

This object indicates the progress of the differential update and is reported back to the installer.

Field Name	**Data Type**	**Init Value**	DescriptionDescr
Current Block	Unsigned int 4 Bytes	0	Current Block that is currently being written to. This is always set to zero for IVI interface since it doesn't apply to IVI.
Total Block Count	Unsigned int 4 Bytes	0	Total number of Blocks in the Diff Package. This is always set to one for IVI diff updates since it is not applicable.
Number of Bytes completed	Unsigned int 4 Bytes	0	Number of Bytes successfully written.
Byte size of current Block	Unsigned int 4 Bytes	0	Total number of Bytes in the Diff block.
Last Written Logical Address	Unsigned int 4 Bytes	0	Last Successfully Written logical Address.

7.1.1.3 FUR-REQ-328493/A-###LSG_DiffUpdater_00003### ProcessDiffPackage_ECU_R

Response code for each Process Next Chunk Response.

Error Code	Block Type	Logical Source Address Present?	Logical Target Address Present?	Data Present?	Record Present in AB Table?	Success
0x00						No Errors
0x01	DCMWE	X	N	X	X	Logical Address required
0x02	DW	X	X	N	X	Data Required
0x03	DCM	X	X	X	N	Unable to Resolve Logical Address.
0x04	M	N	X	X	X	Logical Source Required.
0x05	CME	X	X	Y	X	Unexpected data.
0x06	DCMWE					Diff Package Parse Error, invalid Format.
0xFF						

7.1.1.4 FUR-REQ-328494/A-###LSG_DiffUpdater_00004### Differential Update Trigger

This is the initial call to the Differential Updater that begins the differential update process.

7.1.1.5 FUR-REQ-328496/B-###LSG_DiffUpdater_00006### ProcessNextChunk_SOC

This call causes the differential update state machine to perform a unit of work. The result is returned so that the installer may write the result to its final destination.

Field name	Description
Physical File path	
Differential Update File path	

7.1.1.6 FUR-REQ-328497/A-###LSG_DiffUpdater_00007### Start Diff Update

Trigger for the installer to begin the Diff update.

7.1.1.7 FUR-REQ-328501/A-###LSG_DiffUpdater_00012### Logical Target Address

Logical address of where the result of the differential update is stored. The Installer will use this to resolve it to a physical address.

7.1.1.8 FUR-REQ-328502/A-###LSG_DiffUpdater_00014### Target Size

Size of the target in bytes where the result will be stored.

7.1.1.9 FUR-REQ-328503/A-###LSG_DiffUpdater_00013### Physical Target Address

Actual location where the result of the differential update shall be stored.

7.1.1.10 FUR-REQ-328505/A-###LSG_DiffUpdater_00015### Physical Source Address

The actual source where the diff will be applied to.

7.1.1.11 FUR-REQ-328507/B-###LSG_DiffUpdater_00017### Diff Package

The diff package is the actual data that is applied to the source, that results in the new software written to the output location.

Field	Multiplicity	Description
Number of Blocks	**1**	**Number of Blocks in differential update.**
Diff Block	**1,∞**	**Each block contains:** **Block Type** **logical Target Address/File Path** **Diff Data Size** **DiffData** **Result Checksum** **For IVI Modules :** **LSG_DiffUpdater_00022** **For OVTP ECU's:** **LSG_DiffUpdater_00023**

Byte Layout OVTP ECU's

Diff Package							
1	2	3	4	5	6	7	8
BlockCount (4 Byte)							
Diff Block 1 (n Bytes)							
Diff Block 2 (n Bytes)							
…							
Diff Block N (n Bytes)							

7.1.1.12 FUR-REQ-328508/C-###LSG_DiffUpdater_00023### Diff Block OVTP ECU's

The Diff Block is a single patch that creates an update between 2 contigous regions of memory, this may be achieved by different operations as defined for Blocktype.

Diff Block							
0	1	2	3	4	5	6	7
BlockType (1 Byte) 0 - Diff 1 - Copy 2 - Write 3 – Move							
Logical Source Address (4 Bytes) If Block type is Write, this address should be set to **0x00000000** Required for Move, Copy and Diff.							
Logical Target Address (4 Bytes)							
Logical Target Length **n** (4 Bytes)							
DiffData Size (4 Bytes) **Required** **For Copy and Move, this is set to 0x00000000**							
Diff Data (**n Bytes) Required for Diff and Write operation**							
Checksum CRC16-CITT (2 Bytes)							

7.1.1.13 FUR-REQ-328509/B-###LSG_DiffUpdater_00022### Diff Block for File Based ECU's

Logical description, of a differential software package for IVI modules. The RAW and SPARSE formats should never be passed to the Diff updater.

Field	Description	
Format	RAW	RAW Image or binary to be installed
	SPARSE	Android Sparse Format Image to be installed
	RAW_DIFF	Binary Diff of a RAW Image to be installed. Diff applies to the current image, and will be written to the destination/alternate partition.
Name	File name	
Destination	Logical Filepath that may resolved by the module to a specific location.	
Sha256Hash	Sha256 in base16 text encoding. This hash is calculated over the file referenced in the name field.	

7.1.1.14 FUR-REQ-328510/A-###LSG_DiffUpdater_00018### Logical Target File Path

A string that the module will resolve to an absolute file path to where the result of the differential update will be stored. In many scenerios this shall also be used to resolve the Source File path when applicable.

7.1.1.15 FUR-REQ-328511/A-###LSG_DiffUpdater_00019### Physical Target File Path

The absolute file path of where the result of the differential update shall be stored.

7.1.1.16 FUR-REQ-328512/A-###LSG_DiffUpdater_00020### Physical Source File Path

The absolute file path of where the result of the differential update shall be applied to.

Data Type	Init Value	Default Value (missing signal)

7.1.1.17 FUR-REQ-328513/A-###LSG_DiffUpdater_00021### Diff Block Filepath

Path to the Diff Data File, for intended to be used with a single source and single destination.

Data Type	Init Value	Default Value (missing signal)

7.1.1.18 FUR-REQ-369556/A-###LSG_DiffUpdater_00038### Inititialize

Sends the ECU diff upates params to the diff updater. This is also used for recovering state during a resume from a pause or power loss.

7.1.2 Logical Parameters

#Macro: Add Ins -> Add Requirement macro (select “Logical Parameter” as type)

7.1.2.1 FUR-REQ-328516/B-###LPR_DiffUpdater_00004### ECU Differential Update Package Read Function Pointer

Asychronous issues internally. The diff updater will make the same call each time until its successfull. The Installer will handle retries. The function will return a status of pending, success, and failed.

Parameters:

`[in,out]`	**buf**	Buffer for data read from the storage.
`[in]`	**address**	Position from which data should be read.
`[in]`	**len**	Length of data to be read.
`[in,out]`	**status**	Enum(Success,Pending, Failed)

Return Values:

number of bytes actually read on success,

>0	Number of bytes actually read on success.
0	Read I/O is pending
-1	File of Stream is ended
-2	I/O Failed

.

7.1.2.2 FUR-REQ-328517/A-###LPR_DiffUpdater_00005### Differential Update State NVS

Location where the differential State is stored. This is required for resuming differential installations across power cycles, and persisting any variables required by the Diff Updater. This state is persisted by the installer, and passed back to the diff updater as a parameter during initialization.

7.1.2.3 FUR-REQ-328518/A-###LPR_DiffUpdater_00006### Diff Updater RAM Buffer Length

Length of buffer.

7.1.2.4 FUR-REQ-328519/A-###LPR_DiffUpdater_00007### ECU Source Read Function Pointer

Used by the Diff updater to read the source. This is also an Asynchronous non blocking call. The function referenced by the pointer will handle the Asychronous issues internally. The diff updater will make the same call each time until its successfull. The Installer will handle retries. The function will return a status of pending, success, and failed.

7.1.2.5 FUR-REQ-328520/A-###LPR_DiffUpdater_00008### ECU Init_Mode

Specifies whether the the Differential Update state is to be resumed or initialized.

7.1.2.6 FUR-REQ-328521/A-###LSG_DiffUpdater_00024### Initialize

Message containing all the logica parameters required by the diff update SM.

Field	Data Type	Description
Diff Updater RAM Limit	Int	Max amount of RAM to be used by the Diff Updater
Progress NVS	Pointer	Location where progress of the differential update is stored.
Differential Update NVS	Pointer	Location where the differential package is stored.
Differential Update State NVS	Pointer	Location where any diff update application variables are stored.

7.1.2.7 FUR-REQ-328522/B-###LSG_DiffUpdater_00025### ProcessNextChunk_SOC_R

This is a response indicating the current status of the differential update. (Removed File path references.)

Name	Data Type	Description
Progress	O:Buffer	Buffer pass by reference for storing progress.
Result	Buffer	Buffer where Result is written.

7.1.2.8 FUR-REQ-328523/B-###LSG_DiffUpdater_00026### ProcessNextChunk_ECU

When this message is sent the Diff update SM will perform the next step in the diff update process, if it yields a result it will write it to the final location via call back function.

7.1.2.9 FUR-REQ-328524/A-###LSG_DiffUpdater_00027### SWU Diff Updater RAM Buffer

Configurable buffer that corresponds to max flash write size. This intended for use by the Diff Updater, but is allocated by the IVI installer.

7.1.2.10 FUR-REQ-328525/A-###LSG_DiffUpdater_00028### SWU Diff Updater RAM Buffer Size

Length of the Buffer, which also serves as the RAM limit for the Diff Updater.
32 bit unsigned int

7.1.2.11 FUR-REQ-328527/B-###LSG_DiffUpdater_00030### SWU Differential Update Read Function Pointer

Location where the differential package is stored.
The function will return a status of success, and failed.
Parameters:

[in,out]	***data**	Buffer for data read from the storage.
[in]	**offset**	Position from which data should be read.
[in]	**size**	Length of data to be read.
[in]	***userdata**	readDiffDataFuncUserData

Return Value:

>0	Number of bytes actually read on success
0	0 Read I/O is pending
-1	-1 File or Stream is ended
-2	I/O Failure

7.1.2.12 FUR-REQ-328528/B-###LSG_DiffUpdater_00031### SWU Source Read Function Pointer

Function pointer used for reading from the source on the ECU.
Parameters:

[in,out]	**data**	Buffer for data read from the storage.
[in]	**offset**	Position from which data should be read.
[in]	**size**	Length of data to be read.
[in]	**userdata**	readSourceFuncUserData

Return Values:
Return Value:

>0	Number of bytes actually read on success
0	0 Read I/O is pending
-1	-1 File or Stream is ended
-2	I/O Failure

7.1.2.13 FUR-REQ-328529/A-###LSG_DiffUpdater_00032### readSourceFuncUserData

SWU Agent data for internal use.

Data Type	Init Value	Default Value (missing signal)

void*	Null	Null

7.1.2.14 FUR-REQ-328530/A-###LSG_DiffUpdater_00033### readDiffDataFuncUserData

SWU Agent data for internal use.

Data Type	Init Value	Default Value (missing signal)
void*	Null	Null

7.1.2.15 FUR-REQ-328531/A-###LSG_DiffUpdater_00034### Differential Update State Buffer

Location where the differential State is stored. This is required for resuming differential installations across power cycles. Initialzed to Null by the agent. unless the state is being resumed from a power loss.

7.1.2.16 FUR-REQ-328532/A-###LSG_DiffUpdater_00035### stateDataSize

Size of the state data buffer, this is defined by the Diff updater and is initialized as 0 by the Agent, unless the state is being resumed from a power loss.

7.1.3 Data Types

#Macro: Add Ins -> Add Requirement macro (select “Data Type” as type)

7.2 Attachments