API Documentation for Firmware Update Protocols

Directory: core/mctp/ mctp_interface.h

1. mctp_interface_issue_request(macro)

Encoded into a MCTP type message and send it to command channel.

```
int mctp_interface_issue_request (struct mctp_interface *mctp, struct cmd_channel *channel,
    uint8_t dest_addr, uint8_t dest_eid, uint8_t *request, size_t length, uint8_t *msg_buffer,
    size_t max_length, uint32_t timeout_ms)
```

Parameters

Name	Type	Description
mctp	mctp_interface *	MCTP instance that will be processing the request message.
channel	cmd_channel *	Command channel to use for transmitting the packets.
dest_addr	uint8_t	The destination address for the request.
dest_eid	uint8_t	The destination EID for the request.
request	uint8_t *	Buffer that contains the request body to send.
length	size_t	Length of the request message before any packetization.
msg_buffer	uint8_t *	Buffer that will be used to store the packetized message. This can be overlapping with the request buffer. If the buffers overlap, the request data will be modified upon return.
max_length	size_t	Maximum length of the message buffer.
timeout_ms	uint32_t	Timeout period in milliseconds to wait for response to be received.

Returns

Name (int)	Description

status	return 1 on success

Description

Packetize a request message and send it over a command channel. This call will block until the full message has been transmitted and a response has been received or the operation times out, unless a timeout_ms of 0 is set at which point request is sent and function returns immediately.

Directory: core/cmd_inferface/ cmd_channel.h

1. cmd_channel_receive_and_process (macro)

For Receiving and Processing a MCTP packet.

```
int cmd_channel_receive_and_process (struct cmd_channel *channel, struct mctp_interface
*mctp,
   int ms_timeout)
```

Parameters

Name	Type	Description
channel	cmd_channel *	The channel to receive a packet from.
mctp	mctp_interface *	The MCTP interface to use for processing the received packet.
ms_timeout	int	The amount of time to wait to receive a packet, in milliseconds.

Returns

Name (int)	Description
status	return 1 on success

Description

Receive a single packet from the command channel and process it. Errors will be logged.

Directory: core/pldm/cmd_interface_pldm.h/.c

1. cmd_interface_pldm_process_pldm_protocol_message (macro)

```
static int cmd_interface_pldm_process_pldm_protocol_message (
    struct cmd_interface_pldm *intf, struct cmd_interface_msg *message, uint8_t
*pldm_command)
```

Parameters

Name	Type	Description
intf	struct cmd_interface_pldm *	The command interface that will process the message.
message	struct cmd_interface_msg	The message being processed.
pldm_command	uint8_t	Pointer to hold command ID of incoming message.

Returns

Name (int)	Description
status	return 1 on success. 0 if the message was successfully processed or an error code

Description

* Pre-process received PLDM FWUP protocol message.

2. cmd_interface_pldm_process_request (macro)

```
static int cmd_interface_pldm_process_request (struct cmd_interface *intf, struct
cmd_interface_msg *request)
```

Parameters

Name	Type	Description
intf	struct cmd_interface *	The command interface that will process the request.
request	struct cmd_interface_msg *	The request data to process. This will be updated to contain a response, if necessary.

Returns

Name (int)	Description
status	0 if the request was successfully processed or an error code.

Description

 * Process a PLDM FWUP received request.

3. cmd_interface_pldm_process_response (macro)

```
static int cmd_interface_pldm_process_response (struct cmd_interface *intf, struct

cmd_interface_msg *response)
```

Parameters

Name	Type	Description
intf	struct cmd_interface *	The command interface that will process the request.
response	struct cmd_interface_msg *	The response data to process.

Returns

Name (int)	Description
status	0 if the message was successfully processed or an error code

Description

* Process a PLDM FWUP received response.

4. cmd_interface_pldm_generate_error_packet (macro)

```
static int cmd_interface_pldm_generate_error_packet (struct cmd_interface *intf,
    struct cmd_interface_msg *request, uint8_t error_code, uint32_t error_data, uint8_t
cmd_set)
```

Parameters

Name	Type	Description
intf	struct cmd_interface*	The command interface to utilize.
request	struct cmd_interface_msg *	The request container to utilize.
error_code	uint8_t	Identifier for the error.
error_data	uint32_t	Data for the error condition.
cmd_set	uint8_t	Command set to respond on.

Returns

Name (int)	Description
CMD_HANDLER_PLDM_UNSUPPORTED_OPERATION	CMD_HANDLER_PLDM_UNSUPPORTED_OPERATION

Description

* Generate a message to indicate an error condition.

5. cmd_interface_pldm_init (macro)

```
int cmd_interface_pldm_init (struct cmd_interface_pldm *intf,
    struct pldm_fwup_manager *fwup_mgr, struct device_manager *device_mgr)
```

Parameters

Name	Type	Description
intf	struct cmd_interface_pldm *	The PLDM control command interface instance to initialize.

fwup_mgr	struct pldm_fwup_manager *	The firmware update manager linked to the command interface.
device_mgr	struct device_manager *	The device manager linked to the command interface.

Returns

Name (int)	Description
status	0 if the message was successfully processed or an error code

Description

* Initialize a PLDM command interface instance.

6. cmd_interface_pldm_deinit (macro)

```
void cmd_interface_pldm_deinit (struct cmd_interface_pldm *intf)
```

Parameters

Name	Type	Description
intf	struct cmd_interface_pldm*	The PLDM FWUP command interface instance to deinitialize

Description

* Deinitialize PLDM FWUP command interface instance.

Directory: core/pldm/ pldm_fwup_handeler.h/.c

* A handler for performing a PLDM-based firmware update as definied in DSP0267.
 *
 * The handler contains two internal references to functions that will execute an update when Cerberus is operating as the Update Agent or as the Firmware Device.
 *
 * @note For AMI, the firmware update is performed linearly and does not use the event/periodic task handlers. Parallelization is left up to the AMI team and if used changes will have to be made to the FWUP manager and protocol commands. Also, GetStatus is the only PLDM command not utilized during the update process.

1. pldm_fwup_handler_generate_request (macro)

```
int pldm_fwup_handler_generate_request(struct cmd_interface *intf, int command, uint8_t
*buffer, size_t buf_len)
```

Parameters

Name	Type	Description
intf	struct cmd_interface *	The command interface that will generate the request.
command	int	The PLDM FWUP command
buffer	uint8_t *	The buffer to store the PLDM message
buf_len	size_t	The buffer length.

Returns

Name (int)	Description
status	size of the request or pldm_completion_codes

Description

* Generate PLDM FWUP request for use in the MCTP interface.

2. pldm_fwup_handler_receive_and_respond_full_mctp_message (macro)

```
int pldm_fwup_handler_receive_and_respond_full_mctp_message(struct cmd_channel
*channel, struct mctp_interface *mctp, int timeout_ms)
```

Parameters

Name	Type	Description
channel	struct cmd_channel *	The command channel for sending and receiving packets.
mctp	struct mctp_interface *	The MCTP protocol handler to use for packet processing.
timeout_ms	int	The time to wait in miliseconds for receiving a single packet.

Returns

Name (int)	Description
status	0 if all the packets were received and the message processed otherwise an error code.

Description

* Receive all the packets for a MCTP message, process it, and send back a response.

3. pldm_fwup_handler_send_and_receive_full_mctp_message (macro)

```
int pldm_fwup_handler_send_and_receive_full_mctp_message(struct pldm_fwup_handler
*handler, int command, uint8_t fd_eid, uint8_t fd_addr)
```

Parameters

Name	Type	Description
handler	struct pldm_fwup_handler *	The firmware update handler.
command	int	The PLDM FWUP command
fd_eid	uint8_t *	The endpoint ID of the device to send the message to.
fd_addr	uint8_t	The SMBus address of the firmware device to send the message to.

Returns

Name (int)	Description
status	0 if the MCTP message was sent and a response was processed otherwise an error code.

Description

* Send a full MCTP message, receive a response, and process it.

4. pldm_fwup_handler_check_operation_status (macro)

```
int pldm_fwup_handler_check_operation_status(int transport_status, int protocol_status)
```

Parameters

Name	Type	Description
transport_status	int	The transport layer status of the operation.
protocol_status	int	The protocol layer status of the operation.

Returns

Name (int)	Description
transport_status	0 if the MCTP operation was successful otherwise an error code.

Description

* Check the status of an MCTP operation.

5. pldm_fwup_handler_run_update_ua(macro)

```
int pldm_fwup_handler_run_update_ua(struct pldm_fwup_handler *handler, bool
inventory_cmds, uint8_t fd_eid, uint8_t fd_addr)
```

Parameters

Name	Type	Description
handler	struct pldm_fwup_handler *	The firmware update handler.
inventory_cmds	bool	A flag indicating that inventory commands should be issued to the device being updated.
fd_eid	uint8_t *	The endpoint ID of the device to send the message to.
fd_addr	uint8_t	The SMBus address of the firmware device to send the message to.

Returns

Name (int)	Description
status	0 if the update was successful otherwise an error code.

Description

* Internal reference to the function that will execute a firmware update when Cerberus is operating as the Update Agent.
* This function will update the Firmware Device with the component images contained in the FWUP UA manager.

6. pldm_fwup_handler_start_update_fd (macro)

```
int pldm_fwup_handler_start_update_fd(struct pldm_fwup_handler *handler, uint8_t ua_eid,

uint8_t ua_addr)
```

Parameters

Name	Type	Description
handler	struct pldm_fwup_handler *	The firmware update handler.
ua_eid	uint8_t *	The endpoint ID of the update agent.
ua_addr	uint8_t	The SMBus address of the update agent.

Returns

Name (int)	Description
status	0 if the update was successful otherwise an error code.

Description
* Internal reference to the function that will execute a firmware update when Cerberus is operating as the Update Agent

7. pldm_fwup_handler_init (macro)

```
int pldm_fwup_handler_init(struct pldm_fwup_handler *handler, struct cmd_channel *channel,
struct mctp_interface *mctp, int timeout_ms)
```

Parameters

Name	Type	Description
handler	struct pldm_fwup_handler *	The firmware update handler instance to initialize.
channel	struct cmd_channel *	The command channel for sending and receiving packets.
mctp	struct mctp_interface *	The MCTP protocol handler to use for packet processing.
timeout_ms	int	The time to wait in miliseconds for receiving a single packet.

Returns

Name (int)	Description
status	0 if all the packets were received and the message processed otherwise an error code.

Description
* Initialize a PLDM firmware update handler instance.

8. pldm_fwup_handler_release (macro)

```
void pldm_fwup_handler_release(struct pldm_fwup_handler *handler)
```

Parameters

Name	Type	Description
handler	struct pldm_fwup_handler *	The firmware update handler to release.

Description

* Release the resources used for handling PLDM firmware update.

Directory: core/pldm/ pldm_fwup_protocol_commands.h/.c

/*******************
* Global Helper functions
*******************/

1. print_buffer_data (macro)

```
void print_buffer_data(const uint8_t *data, size_t len)
```

Parameters

Name	Type	Description
data	const uint8_t *	Buffer to store the data.
len	size_t	Size of the data to be printed.

Description
* Global Helper functions

/*******************
* FD Helper functions
*******************/

2. switch_state(macro)

```
void switch_state(struct pldm_fwup_fd_state *state, enum pldm_firmware_device_states
new_state)
```

Parameters

Name	Type	Description
state	pldm_fwup_fd_state *	Variable context for a PLDM FWUP protocol command interface.
new_state	enum pldm_firmware_device_states	For new PLDM FWUP states.

Description
* FD Helper functions

/******************* * FD Inventory commands *******************/

3. pldm_fwup_process_query_device_identifiers_request (macro)

```
int pldm_fwup_process_query_device_identifiers_request(struct pldm_fwup_fd_state *state,
struct device_manager *device_mgr, struct cmd_interface_msg *request)
```

Parameters

Name	Type	Description
state	struct pldm_fwup_fd_state *	Variable context for a PLDM FWUP.
device_mgr	struct device_manager *	The device manager linked to command interface.
request	struct cmd_interface_msg *	The request data to process. This will be updated to contain a response.

Returns

Name (int)	Description
status	0 on success or an error code.

Description

* Process a QueryDeviceIdentifiers request.
* @note A QueryDeviceIdentifiers request does not contain payload data.

4. pldm_fwup_process_get_firmware_parameters_request (macro)

```
int pldm_fwup_prcocess_get_firmware_parameters_request(struct pldm_fwup_fd_state *state,
struct pldm_fwup_protocol_firmware_parameters *fw_parameters, struct cmd_interface_msg
*request)
```

Parameters

Name	Type	Description
state	struct pldm_fwup_fd_state *	Variable context for a PLDM FWUP.
fw_parameters	struct pldm_fwup_protocol_firmware_parameters *	FD firmware parameters.
request	struct cmd_interface_msg *	The request data to process. This will be updated to contain a response.

Returns

Name (int)	Description
status	0 on success or an error code.

Description
* Process a GetFirmwareParameters request.

/*******************

* FD Update commands

*******************/

5. pldm_fwup_process_request_update_request (macro)

```
int pldm_fwup_process_request_update_request(struct pldm_fwup_fd_state *state, struct
pldm_fwup_flash_manager *flash_mgr, struct pldm_fwup_fd_update_info *update_info, struct
cmd_interface_msg *request)
```

Parameters

Name	Type	Description
state	struct pldm_fwup_fd_state *	Variable context for a PLDM FWUP.
flash_mgr	struct pldm_fwup_flash_manager *	The flash manager for a PLDM FWUP.
update_info	pldm_fwup_fd_update_info *	Update information retained by FD.
request	struct cmd_interface_msg *	The request data to process. This will be updated to contain a response.

Returns

Name (int)	Description
status	0 on success or an error code.

Description

* Process a RequestUpdate request.
* @note For AMI, the function does not handle UNABLE_TO_INITIATE_UPDATE and RETRY_REQUEST_UPDATE errors and corresponding state changes.
 * This should be implemented later on according to some other Cerberus criteria.

6. pldm_fwup_generate_get_package_data_request (macro)

```
int pldm_fwup_generate_get_package_data_request(struct pldm_fwup_fd_state *state,struct
pldm_fwup_protocol_multipart_transfer *get_cmd_state, uint8_t *buffer, size_t buf_len)
```

Parameters

Name	Type	Description
state	struct pldm_fwup_fd_state *	Variable context for a PLDM FWUP.
get_cmd_state	struct pldm_fwup_protocol_multipart_transfer *	Variable context for a multipart transfer.
buffer	uint8_t *	The buffer to contain the request data.
buf_len	size_t	The buffer length.

Returns

Name (int)	Description
status	0 if the request was successfully generated or an error code.

Description

* Generate a GetPackageData request.

7. pldm_fwup_process_get_package_data_response (macro)

```
int pldm_fwup_process_get_package_data_response(struct pldm_fwup_fd_state *state, struct
pldm_fwup_flash_manager *flash_mgr, struct pldm_fwup_protocol_multipart_transfer
*get_cmd_state, struct cmd_interface_msg *response)
```

Parameters

Name	Type	Description
state	struct pldm_fwup_fd_state *	Variable context for a PLDM FWUP.
flash_mgr	struct pldm_fwup_flash_manager *	The flash manager for a PLDM FWUP.
get_cmd_state	struct pldm_fwup_protocol_multipart_transfer *	Variable context for a multipart transfer.
response	struct cmd_interface_msg *	The response data to process.

Returns

Name (int)	Description
status	0 if the response was successfully processed or an error code.

Description

* Process a GetPackageData response.

8. pldm_fwup_process_get_device_meta_data_request (macro)

```
int pldm_fwup_process_get_device_meta_data_request(struct pldm_fwup_fd_state *state,
struct pldm_fwup_flash_manager *flash_mgr, struct pldm_fwup_protocol_multipart_transfer
*get_cmd_state, struct pldm_fwup_fd_update_info *update_info, struct cmd_interface_msg
*request)
```

Parameters

Name	Type	Description
state	struct pldm_fwup_fd_state *	Variable context for a PLDM FWUP.
flash_mgr	struct pldm_fwup_flash_manager *	The flash manager for a PLDM FWUP.
get_cmd_state	struct pldm_fwup_protocol_multipart_transfer *	Variable context for a multipart transfer.
update_info	struct pldm_fwup_fd_update_info *	Update information retained by FD.
request	struct cmd_interface_msg *	The request data to process. This will be updated to contain a response.

Returns

Name (int)	Description
status	0 if the request was successfully processed and a request was generated or an error code.

Description
* Process a GetDeviceMetaData request and generate a response.

9. pldm_fwup_process_pass_component_table_request (macro)

```
int pldm_fwup_process_pass_component_table_request(struct pldm_fwup_fd_state *state,
struct pldm_fwup_fd_update_info *update_info, struct
pldm_fwup_protocol_firmware_parameters *fw_parameters, struct cmd_interface_msg
*request)
```

Parameters

Name	Type	Description
state	struct pldm_fwup_fd_state *	Variable context for a PLDM FWUP.
update_info	struct pldm_fwup_fd_update_info *	Update information retained by FD.
fw_parameters	struct pldm_fwup_protocol_firmware_parameters *	FD firmware parameters.
request	struct cmd_interface_msg *	The request data to process. This will be updated to contain a response.

Returns

Name (int)	Description
status	0 if the request was successfully processed and a request was generated or an error code.

Description

* Process a PassComponentTable request and generate a response.
* @note For AMI, not every component response code is handled since some depend on other components of Cerberus.

10. pldm_fwup_process_update_component_request (macro)

```
int pldm_fwup_process_update_component_request(struct pldm_fwup_fd_state *state,
    struct pldm_fwup_fd_update_info *update_info, struct
pldm_fwup_protocol_component_entry *comp_entries,
    struct cmd_interface_msg *request)
```

Parameters

Name	Type	Description
state	struct pldm_fwup_fd_state *	Variable context for a PLDM FWUP.
update_info	struct pldm_fwup_fd_update_info *	Update information retained by FD.
comp_entries	struct pldm_fwup_protocol_component_entry *	FD firmware parameters.
request	struct cmd_interface_msg *	The request data to process. This will be updated to contain a response.

Returns

Name (int)	Description
status	0 if the request was successfully processed and a request was generated or an error code.

Description

* Process a UpdateComponent request and generate a response.

* @note For AMI, not every component response code is handled since some depend on the external state/status of Cerberus. Also the FD enabled update options flags is simply set to what the UA requested without any additional checks, configuration, etc. and the time before RequestFirmwareData field is set to a dummy value of no significance.

11. pldm_fwup_generate_request_firmware_data_request (macro)

```
int pldm_fwup_generate_request_firmware_data_request(struct pldm_fwup_fd_state *state,

struct pldm_fwup_fd_update_info *update_info, uint8_t *buffer, size_t buf_len)
```

Parameters

Name	Type	Description
state	struct pldm_fwup_fd_state *	Variable context for a PLDM FWUP.
update_info	struct pldm_fwup_fd_update_info *	Update information retained by FD.
buffer	uint8_t *	The buffer to contain the request data.
buf_len	size_t	The buffer length.

Returns

Name (int)	Description
status	0 if the request was successfully generated or an error code.

Description

* Generate a RequestFirmwareData request.

12. pldm_fwup_process_request_firmware_data_response (macro)

```
int pldm_fwup_process_request_firmware_data_response(struct pldm_fwup_fd_state *state,
   struct pldm_fwup_fd_update_info *update_info, struct pldm_fwup_flash_manager
*flash_mgr,
   struct cmd_interface_msg *response)
```

Parameters

Name	Type	Description
state	struct pldm_fwup_fd_state *	Variable context for a PLDM FWUP.
update_info	struct pldm_fwup_fd_update_info *	Update information retained by FD.
flash_mgr	struct pldm_fwup_flash_manager *	The flash manager for a PLDM FWUP.
response	struct cmd_interface_msg *	The response data to process.

Returns

Name (int)	Description
status	0 if the response was successfully processed or an error code.

Description
* Process a RequestFirmwareData response.

13. pldm_fwup_generate_transfer_complete_request (macro)

```
int pldm_fwup_generate_transfer_complete_request(struct pldm_fwup_fd_state *state, uint8_t
*buffer, size_t buf_len)
```

Parameters

Name	Type	Description
state	struct pldm_fwup_fd_state *	Variable context for a PLDM FWUP.
buffer	uint8_t *	The buffer to contain the request data.
buf_len	size_t	The buffer length.

Returns

Name (int)	Description
status	0 if the request was successfully generated or an error code.

Description
* Generate a TransferComplete request.
* @note For AMI, the transfer result is based on what the last completion code received during RequestFirmwareData was set to.
* Checks for all transfer results should be implemented based on specific needs.

14. pldm_fwup_process_transfer_complete_response (macro)

```
int pldm_fwup_process_transfer_complete_response(struct pldm_fwup_fd_state *state, struct
cmd_interface_msg *response)
```

Parameters

Name	Type	Description
state	struct pldm_fwup_fd_state *	Variable context for a PLDM FWUP.
response	struct cmd_interface_msg *	The response data to process.

Returns

Name (int)	Description
status	0 if the response was successfully processed or an error code.

Description
* Process a TransferComplete response.

15. pldm_fwup_generate_verify_complete_request (macro)

```
int pldm_fwup_generate_verify_complete_request(struct pldm_fwup_fd_state *state, uint8_t
*buffer, size_t buf_len)
```

Parameters

Name	Type	Description
state	struct pldm_fwup_fd_state *	Variable context for a PLDM FWUP.
buffer	uint8_t *	The buffer to contain the request data.
buf_len	size_t	The buffer length.

Returns

Name (int)	Description

status	0 if the request was successfully generated or an error code.

Description
* Generate a VerifyComplete request.
* @note For AMI, this is skeleton code. Verification of the requested firmware image is left up to the AMI team and their specific requirements.

16. pldm_fwup_process_verify_complete_response (macro)

```
int pldm_fwup_process_verify_complete_response(struct pldm_fwup_fd_state *state, struct
cmd_interface_msg *response)
```

Parameters

Name	Type	Description
state	struct pldm_fwup_fd_state *	Variable context for a PLDM FWUP.
response	struct cmd_interface_msg *	The response data to process.

Returns

Name (int)	Description
status	0 if the response was successfully processed or an error code.

Description
* Process a VerifyComplete response.

17. pldm_fwup_generate_apply_complete_request (macro)

```
int pldm_fwup_generate_apply_complete_request(struct pldm_fwup_fd_state *state, uint8_t
*buffer, size_t buf_len)
```

Parameters

Name	Type	Description

state	struct pldm_fwup_fd_state *	Variable context for a PLDM FWUP.
buffer	uint8_t *	The buffer to contain the request data.
buf_len	size_t	The buffer length.

Returns

Name (int)	Description
status	0 if the request was successfully generated or an error code.

Description

* Generate a ApplyComplete request.

* @note For AMI, this is skeleton code. DMTF specifies that during RequestFirmwareData the received firmware image is stored in volatile memory. However, the RequestFirmwareData API immediately writes to a designated flash region. This means that ApplyComplete can either be skipped or the flash region can be treated as temporary and ApplyComplete could transfer to another more permanent region.

18. pldm_fwup_process_apply_complete_response (macro)

```
int pldm_fwup_process_apply_complete_response(struct pldm_fwup_fd_state *state, struct

cmd_interface_msg *response)
```

Parameters

Name	Type	Description
state	struct pldm_fwup_fd_state *	Variable context for a PLDM FWUP.
response	struct cmd_interface_msg *	The response data to process.

Returns

Name (int)	Description
status	0 if the response was successfully processed or an error code.

Description

* Process a ApplyComplete response.

19. pldm_fwup_generate_get_meta_data_request (macro)

```
int pldm_fwup_generate_get_meta_data_request(struct pldm_fwup_fd_state *state, struct
pldm_fwup_protocol_multipart_transfer *get_cmd_state, uint8_t *buffer, size_t buf_len)
```

Parameters

Name	Type	Description
state	struct pldm_fwup_fd_state *	Variable context for a PLDM FWUP.
get_cmd_state	struct pldm_fwup_protocol_multipart_transfer *	Variable context for a multipart transfer.
buffer	uint8_t *	The buffer to contain the request data.
buf_len	size_t	The buffer length.

Returns

Name (int)	Description
status	0 if the request was successfully generated or an error code.

Description
* Generate a GetMetaData request.

20. pldm_fwup_process_get_meta_data_response (macro)

```
int pldm_fwup_process_get_meta_data_response(struct pldm_fwup_fd_state *state,
   struct pldm_fwup_flash_manager *flash_mgr, struct pldm_fwup_protocol_multipart_transfer
*get_cmd_state,
   struct cmd_interface_msg *response)
```

Parameters

Name	Type	Description

state	struct pldm_fwup_fd_state *	Variable context for a PLDM FWUP.
flash_mgr	struct pldm_fwup_flash_manager *	The flash manager for a PLDM FWUP.
get_cmd_state	struct pldm_fwup_protocol_multipart_transfer *	Variable context for a multipart transfer.
response	struct cmd_interface_msg *	The response data to process.

Returns

Name (int)	Description
status	0 if the response was successfully processed or an error code.

Description

* Process a GetMetaData response.

21. pldm_fwup_process_activate_firmware_request (macro)

```
int pldm_fwup_process_activate_firmware_request(struct pldm_fwup_fd_state *state,

   struct pldm_fwup_fd_update_info *update_info, struct cmd_interface_msg *request)
```

Parameters

Name	Type	Description
state	struct pldm_fwup_fd_state *	Variable context for a PLDM FWUP.
update_info	struct pldm_fwup_fd_update_info *	Update information retained by FD.
request	struct cmd_interface_msg *	The request data to process. This will be updated to contain a response.

Returns

Name (int)	Description
status	0 on success or an error code.

Description
* Process a ActivateFirmware request.
* @note For AMI, this is skeleton code. The handling of the self contained activation request, the assignment of the estimated time for activation, determining whether an incomplete update has occurred, and determining is self contained activation is supported is left to the AMI team.

pldm_fwup_process_get_status_request (macro)

```
int pldm_fwup_process_get_status_request(struct pldm_fwup_fd_state *state, struct
pldm_fwup_fd_update_info *update_info, struct cmd_interface_msg *request)
```

Parameters

Name	Type	Description
state	struct pldm_fwup_fd_state *	Variable context for a PLDM FWUP.
update_info	struct pldm_fwup_fd_update_info *	Update information retained by FD.
request	struct cmd_interface_msg *	The request data to process. This will be updated to contain a response.

Returns

Name (int)	Description
status	0 on success or an error code.

Description
* Process a GetStatus request.
* @note For AMI, assumes progress percent is not supported and timeouts are not checked for the aux state status and reason code.

22. pldm_fwup_process_cancel_update_component_request (macro)

```
int pldm_fwup_process_cancel_update_component_request(struct pldm_fwup_fd_state *state,
struct pldm_fwup_fd_update_info *update_info, struct cmd_interface_msg *request)
```

Parameters

Name	Type	Description
state	struct pldm_fwup_fd_state *	Variable context for a PLDM FWUP.
update_info	struct pldm_fwup_fd_update_info *	Update information retained by FD.
request	struct cmd_interface_msg *	The request data to process. This will be updated to contain a response.

Returns

Name (int)	Description
status	0 on success or an error code.

Description

* Process a CancelUpdateComponent request.
* @note For AMI, the busy in background completion code is not checked and will be left to the AMI team.

23. pldm_fwup_process_cancel_update_request (macro)

```
int pldm_fwup_process_cancel_update_request(struct pldm_fwup_fd_state *state,
    struct pldm_fwup_fd_update_info *update_info, struct pldm_fwup_flash_manager
*flash_mgr, struct cmd_interface_msg *request)
```

Parameters

Name	Type	Description

state	struct pldm_fwup_fd_state *	Variable context for a PLDM FWUP.
update_info	struct pldm_fwup_fd_update_info *	Update information retained by FD.
flash_mgr	struct pldm_fwup_flash_manager *	The flash manager for a PLDM FWUP.
request	struct cmd_interface_msg *	The request data to process. This will be updated to contain a response.

Returns

Name (int)	Description
status	0 on success or an error code.

Description

* Process a CancelUpdate request.

* @note For AMI, this is skeleton code. Some reset is done upon receiving a CancelUpdate command, but additional implementation is left to the AMI team.

 * The assignment of the non functioning component indication and bitmap and the busy in background completion code is left to the AMI team.

/*******************

* UA Inventory commands ********************/

24. pldm_fwup_generate_query_device_identifiers_request (macro)

```
int pldm_fwup_generate_query_device_identifiers_request(struct pldm_fwup_ua_state *state,
uint8_t *buffer, size_t buf_len)
```

Parameters

Name	Type	Description
state	struct pldm_fwup_ua_state *	Variable context for a PLDM FWUP.
buffer	uint8_t *	The buffer to contain the request data.
buf_len	size_t	The buffer length.

Returns

Name (int)	Description
status	size of the message payload or an error code.

Description

* Generate a QueryDeviceIdentifiers request.

25. pldm_fwup_process_query_device_identifiers_response (macro)

```
int pldm_fwup_process_query_device_identifiers_response(struct pldm_fwup_ua_state *state,
struct device_manager *device_mgr, struct cmd_interface_msg *response)
```

Parameters

Name	Type	Description

state	struct pldm_fwup_ua_state *	Variable context for a PLDM FWUP.
device_mgr	struct device_manager *	The device manager linked to command interface.
response	struct cmd_interface_msg *	The response data to process.

Returns

Name (int)	Description
status	0 on success or an error code.

Description
* Generate a QueryDeviceIdentifiers response.

26. pldm_fwup_generate_get_firmware_parameters_request (macro)

```
int pldm_fwup_generate_get_firmware_parameters_request(struct pldm_fwup_ua_state
*state, uint8_t *buffer, size_t buf_len)
```

Parameters

Name	Type	Description
state	struct pldm_fwup_ua_state *	Variable context for a PLDM FWUP.
buffer	uint8_t *	The buffer to contain the request data.
buf_len	size_t	The buffer length.

Returns

Name (int)	Description
status	size of the message payload or an error code.

Description
* Generate a GetFirmwareParameters request.

27. pldm_fwup_process_query_device_identifiers_response (macro)

```
int pldm_fwup_process_get_firmware_parameters_response(struct pldm_fwup_ua_state
*state, struct pldm_fwup_protocol_firmware_parameters *rec_fw_parameters, struct
cmd_interface_msg *response)
```

Parameters

Name	Type	Description
state	struct pldm_fwup_ua_state *	Variable context for a PLDM FWUP.
rec_fw_parameters	pldm_fwup_protocol_firmware_parameters *	Pointer to the firmware parameters received from the FD.
response	struct cmd_interface_msg *	The response data to process.

Returns

Name (int)	Description
status	0 on success or an error code.

Description

* Process a GetFirmwareParameters response.

/*******************
* UA Inventory commands
*******************/

28. pldm_fwup_generate_request_update_request (macro)

```
int pldm_fwup_generate_request_update_request(struct pldm_fwup_ua_manager *ua_mgr,
uint8_t *buffer, size_t buf_len)
```

Parameters

Name	Type	Description
ua_mgr	struct pldm_fwup_ua_manager *	The manager for the UA.
buffer	uint8_t *	The buffer to contain the request data.
buf_len	size_t	The buffer length.

Returns

Name (int)	Description
status	size of the message payload or an error code.

Description

* Generate a RequestUpdate request.

29. pldm_fwup_process_request_update_response (macro)

```
int pldm_fwup_process_request_update_response(struct pldm_fwup_ua_state *state,
    struct pldm_fwup_ua_update_info *update_info, struct cmd_interface_msg *response)
```

Parameters

Name	Type	Description
state	struct pldm_fwup_ua_state *	Variable context for a PLDM FWUP.
update_info	struct pldm_fwup_ua_update_info *	Update information retained by UA.
response	struct cmd_interface_msg *	The response data to process.

Returns

Name (int)	Description
status	0 on success or an error code.

Description

* Process a RequestUpdate response.

30. pldm_fwup_process_get_package_data_request (macro)

```
int pldm_fwup_process_get_package_data_request(struct pldm_fwup_ua_state *state,
   struct pldm_fwup_flash_manager *flash_mgr, struct pldm_fwup_protocol_multipart_transfer
*get_cmd_state,
   struct cmd_interface_msg *request)
```

Parameters

Name	Type	Description
state	struct pldm_fwup_ua_state *	Variable context for a PLDM FWUP.
flash_mgr	struct pldm_fwup_flash_manager *	The flash manager for a PLDM FWUP.
get_cmd_state	struct pldm_fwup_protocol_multipart_transfer *	Variable context for a multipart transfer.
request	struct cmd_interface_msg *	The request data to process. This will be updated to contain a response.

Returns

Name (int)	Description
status	0 if the request was successfully processed and a request was generated or an error code.

Description

* Process a GetPackageData request and generate a response.

31. pldm_fwup_generate_get_device_meta_data_request (macro)

```
int pldm_fwup_generate_get_device_meta_data_request(struct pldm_fwup_ua_state *state,
struct pldm_fwup_protocol_multipart_transfer *get_cmd_state,
    uint8_t *buffer, size_t buf_len)
```

Parameters

Name	Type	Description
state	struct pldm_fwup_ua_state *	Variable context for a PLDM FWUP.
get_cmd_state	struct pldm_fwup_protocol_multipart_transfer *	Variable context for a multipart transfer.
buffer	uint8_t *	The buffer to contain the request data.
buf_len	size_t	The buffer length.

Returns

Name (int)	Description
status	0 if the request was successfully generated or an error code.

Description
* Generate a GetDeviceMetaData request.

32. pldm_fwup_process_get_device_meta_data_response (macro)

```
int pldm_fwup_process_get_device_meta_data_response(struct pldm_fwup_ua_state *state,
    struct pldm_fwup_flash_manager *flash_mgr, struct pldm_fwup_protocol_multipart_transfer
*get_cmd_state,
    struct cmd_interface_msg *response)
```

Parameters

Name	Type	Description
state	struct pldm_fwup_ua_state *	Variable context for a PLDM FWUP.
flash_mgr	struct pldm_fwup_flash_manager *	The flash manager for a PLDM FWUP.
get_cmd_state	struct pldm_fwup_protocol_multipart_transfer *	Variable context for a multipart transfer.
response	struct cmd_interface_msg *	The response data to process.

Returns

Name (int)	Description
status	0 if the response was successfully processed or an error code.

Description

* Process a GetDeviceMetaData response.

33. pldm_fwup_generate_pass_component_table_request (macro)

```
int pldm_fwup_generate_pass_component_table_request(struct pldm_fwup_ua_manager
*ua_mgr, uint8_t *buffer, size_t buf_len)
```

Parameters

Name	Type	Description
ua_mgr	struct pldm_fwup_ua_manager *	The manager for the UA.
buffer	uint8_t *	The buffer to contain the request data.
buf_len	size_t	The buffer length.

Returns

Name (int)	Description
status	0 if the request was successfully generated or an error code.

Description
* Generate a PassComponentTable request.

34. pldm_fwup_process_pass_component_table_response (macro)

```
int pldm_fwup_process_pass_component_table_response(struct pldm_fwup_ua_state *state,
   struct pldm_fwup_ua_update_info *update_info, struct cmd_interface_msg *response)
```

Parameters

Name	Type	Description
state	struct pldm_fwup_ua_state *	Variable context for a PLDM FWUP.
update_info	struct pldm_fwup_ua_update_info *	Update information retained by UA.
response	struct cmd_interface_msg *	The response data to process.

Returns

Name (int)	Description
status	0 if the response was successfully processed or an error code.

Description
* Process a PassComponentTable response.

35. pldm_fwup_generate_update_component_request(macro)

```
int pldm_fwup_generate_update_component_request(struct pldm_fwup_ua_state *state,
    uint16_t current_comp_num, struct pldm_fwup_fup_component_image_entry
*comp_img_entries,
    struct pldm_fwup_protocol_firmware_parameters *rec_fw_parameters, uint8_t *buffer,
size_t buf_len)
```

Parameters

Name	Type	Description
state	struct pldm_fwup_ua_state *	Variable context for a PLDM FWUP.
current_comp_num	uint16_t	Current component to be updated.
comp_img_entries	struct pldm_fwup_fup_component_image_entry *	Component image information entries from the FUP.
rec_fw_parameters	struct pldm_fwup_protocol_firmware_parameters *	Received FD firmware parameters.
buffer	uint8_t *	The buffer to contain the request data.
buf_len	size_t	The buffer length.

Returns

Name (int)	Description
status	0 if the request was successfully generated or an error code.

Description

* Generate a UpdateComponent request.

36. pldm_fwup_process_update_component_response (macro)

```
int pldm_fwup_process_update_component_response(struct pldm_fwup_ua_state *state,
    struct pldm_fwup_ua_update_info *update_info, struct cmd_interface_msg *response)
```

Parameters

Name	Type	Description
state	struct pldm_fwup_ua_state *	Variable context for a PLDM FWUP.
update_info	struct pldm_fwup_ua_update_info *	Update information retained by UA.
response	struct cmd_interface_msg *	The response data to process.

Returns

Name (int)	Description
status	0 if the response was successfully processed or an error code.

Description

* Process a UpdateComponent response.

37. pldm_fwup_generate_update_component_request(macro)

```
int pldm_fwup_process_request_firmware_data_request(struct pldm_fwup_ua_state *state,
    uint16_t current_comp_num, struct pldm_fwup_fup_component_image_entry
*comp_img_entries,
    struct pldm_fwup_flash_manager *flash_mgr, struct cmd_interface_msg *request)
```

Parameters

Name	Type	Description
state	struct pldm_fwup_ua_state *	Variable context for a PLDM FWUP.
current_comp_num	uint16_t	Current component to be updated.
comp_img_entries	struct pldm_fwup_fup_component_image_entry *	Component image information entries from the FUP.
flash_mgr	struct pldm_fwup_flash_manager *	The flash manager for a PLDM FWUP.
request	struct cmd_interface_msg *	The request data to process. This will be updated to contain a response.

Returns

Name (int)	Description
status	0 if the request was successfully generated or an error code.

Description

* Process a RequestFirmwareData request and generate a response.

* @note For AMI, RETRY_REQUEST_FW_DATA is returned if the flash region has a length set to zero. This should be changed to a more robust check in the future.

38. pldm_fwup_process_transfer_complete_request (macro)

```
int pldm_fwup_process_transfer_complete_request(struct pldm_fwup_ua_state *state,
   struct pldm_fwup_ua_update_info *update_info, struct cmd_interface_msg *request)
```

Parameters

Name	Type	Description

state	struct pldm_fwup_ua_state *	Variable context for a PLDM FWUP.
update_info	struct pldm_fwup_ua_update_info *	Update information retained by UA.
request	struct cmd_interface_msg *	The request data to process. This will be updated to contain a response.

Returns

Name (int)	Description
status	0 if the request was successfully processed and a request was generated or an error code.

Description
* Process a TransferComplete request.

39. pldm_fwup_process_verify_complete_request (macro)

```
int pldm_fwup_process_verify_complete_request(struct pldm_fwup_ua_state *state,
    struct pldm_fwup_ua_update_info *update_info, struct cmd_interface_msg *request)
```

Parameters

Name	Type	Description
state	struct pldm_fwup_ua_state *	Variable context for a PLDM FWUP.

update_info	struct pldm_fwup_ua_update_info *	Update information retained by UA.
request	struct cmd_interface_msg *	The request data to process. This will be updated to contain a response.

Returns

Name (int)	Description
status	0 if the request was successfully processed and a request was generated or an error code.

Description

* Process a VerifyComplete request.

40. pldm_fwup_process_apply_complete_request (macro)

```
int pldm_fwup_process_apply_complete_request(struct pldm_fwup_ua_state *state,
    struct pldm_fwup_ua_update_info *update_info, struct cmd_interface_msg *request)
```

Parameters

Name	Type	Description
state	struct pldm_fwup_ua_state *	Variable context for a PLDM FWUP.
update_info	struct pldm_fwup_ua_update_info *	Update information retained by UA.
request	struct cmd_interface_msg *	The request data to process. This will be updated to contain a response.

Returns

Name (int)	Description
status	0 if the request was successfully processed and a request was generated or an error code.

Description
* Process a ApplyComplete request.

41. pldm_fwup_process_get_meta_data_request (macro)

```
int pldm_fwup_process_get_meta_data_request(struct pldm_fwup_ua_state *state,
    struct pldm_fwup_flash_manager *flash_mgr, struct pldm_fwup_protocol_multipart_transfer
*get_cmd_state,
    struct cmd_interface_msg *request)
```

Parameters

Name	Type	Description
state	struct pldm_fwup_ua_state *	Variable context for a PLDM FWUP.
flash_mgr	struct pldm_fwup_flash_manager *	The flash manager for a PLDM FWUP.
get_cmd_state	struct pldm_fwup_protocol_multipart_transfer *	Variable context for a multipart transfer.
request	struct cmd_interface_msg *	The request data to process. This will be updated to contain a response.

Returns

Name (int)	Description
status	0 if the request was successfully processed and a request was generated or an error code.

Description
* Process a GetMetaData request and generate a response.

42. pldm_fwup_generate_activate_firmware_request (macro)

```
int pldm_fwup_generate_activate_firmware_request(struct pldm_fwup_ua_state *state, uint8_t
*buffer, size_t buf_len)
```

Parameters

Name	Type	Description
state	struct pldm_fwup_ua_state *	Variable context for a PLDM FWUP.
buffer	uint8_t *	The buffer to contain the request data.
buf_len	size_t	The buffer length.

Returns

Name (int)	Description
status	0 if the request was successfully generated or an error code.

Description

* Generate a ActivateFirmware request.

* @note For AMI, this is skeleton code. The activation request field should be set based on the specifics required by the AMI team.

43. pldm_fwup_process_activate_firmware_response (macro)

```
int pldm_fwup_process_activate_firmware_response(struct pldm_fwup_ua_state *state,
   struct pldm_fwup_ua_update_info *update_info, struct cmd_interface_msg *response)
```

Parameters

Name	Type	Description
state	struct pldm_fwup_ua_state *	Variable context for a PLDM FWUP.

update_info	struct pldm_fwup_ua_update_info *	Update information retained by UA.
response	struct cmd_interface_msg *	The response data to process.

Returns

Name (int)	Description
status	0 if the response was successfully processed or an error code.

Description
* Process a ActivateFirmware response.

44. pldm_fwup_generate_get_status_request (macro)

```
int pldm_fwup_generate_get_status_request(struct pldm_fwup_ua_state *state, uint8_t
*buffer, size_t buf_len)
```

Parameters

Name	Type	Description
state	struct pldm_fwup_ua_state *	Variable context for a PLDM FWUP.
buffer	uint8_t *	The buffer to contain the request data.
buf_len	size_t	The buffer length.

Returns

Name (int)	Description
status	size of the message payload or an error code.

Description
* Generate a GetStatus request.

45. pldm_fwup_process_get_status_response (macro)

```
int pldm_fwup_process_get_status_response(struct pldm_fwup_ua_state *state, struct
cmd_interface_msg *response)
```

Parameters

Name	Type	Description
state	struct pldm_fwup_ua_state *	Variable context for a PLDM FWUP.
response	struct cmd_interface_msg *	The response data to process.

Returns

Name (int)	Description
status	0 on success or an error code.

Description

* Process a GetStatus response.
* @note For AMI, this is skeleton code. The current implementation of the firmware update flow does not ever call GetStatus, so the extracted fields are not saved.

46. pldm_fwup_generate_cancel_update_component_request (macro)

```
int pldm_fwup_generate_cancel_update_component_request(struct pldm_fwup_ua_state
*state, uint8_t *buffer, size_t buf_len)
```

Parameters

Name	Type	Description
state	struct pldm_fwup_ua_state *	Variable context for a PLDM FWUP.
buffer	uint8_t *	The buffer to contain the request data.

buf_len	size_t	The buffer length.

Returns

Name (int)	Description
status	size of the message payload or an error code.

Description

* Generate a CancelUpdateComponent request.

47. pldm_fwup_process_cancel_update_component_response (macro)

```
int pldm_fwup_process_cancel_update_component_response(struct pldm_fwup_ua_state
*state, struct cmd_interface_msg *response)
```

Parameters

Name	Type	Description
state	struct pldm_fwup_ua_state *	Variable context for a PLDM FWUP.
response	, struct cmd_interface_msg *	The response data to process.

Returns

Name (int)	Description
status	0 on success or an error code.

Description

* Process a CancelUpdateComponent response.

48. pldm_fwup_generate_cancel_update_request (macro)

```
int pldm_fwup_generate_cancel_update_request(struct pldm_fwup_ua_state *state, uint8_t
*buffer, size_t buf_len)
```

Parameters

Name	Type	Description
state	struct pldm_fwup_ua_state *	Variable context for a PLDM FWUP.
buffer	uint8_t *	The buffer to contain the request data.
buf_len	size_t	The buffer length.

Returns

Name (int)	Description
status	size of the message payload or an error code.

Description
* Generate a CancelUpdate request.

49. pldm_fwup_process_cancel_update_response (macro)

```
int pldm_fwup_process_cancel_update_response(struct pldm_fwup_ua_state *state, struct
cmd_interface_msg *response)
```

Parameters

Name	Type	Description
state	struct pldm_fwup_ua_state *	Variable context for a PLDM FWUP.
response	, struct cmd_interface_msg *	The response data to process.

Returns

Name (int)	Description
status	0 on success or an error code.

Description

* Process a CancelUpdate response.
* @note For AMI, this is skeleton code. The handling of the non functioning component indication and bitmap is left to the AMI tea

Directory: core/pldm/ pldm_fwup_manager.h/.c

1. pldm_fwup_manager_init (macro)

```
int pldm_fwup_manager_init(struct pldm_fwup_manager *fwup_mgr,
    struct pldm_fwup_protocol_firmware_parameters *fd_fw_parameters,  struct
pldm_fwup_fup_component_image_entry *fup_comp_img_list,
    struct pldm_fwup_flash_manager *fd_flash_mgr, struct pldm_fwup_flash_manager
*ua_flash_mgr,
    struct pldm_fwup_protocol_version_string *fup_comp_img_set_ver, uint16_t
num_components)
```

Parameters

Name	Type	Description
fwup_mgr	struct pldm_fwup_manager *fwup	The PLDM FWUP manager instance to initialize.
fd_fw_parameters	struct pldm_fwup_protocol_firmware_parameters *	The FD firmware parameters
fup_comp_img_list	struct pldm_fwup_fup_component_image_entry *	The FUP component image information list
fd_flash_mgr	struct pldm_fwup_flash_manager *	The FWUP flash manager for the FD
ua_flash_mgr	struct pldm_fwup_flash_manager *	The FWUP flash manager for the UA
fup_comp_img_set_ver	struct pldm_fwup_protocol_version_string *	The component image set obtained from the FUP.
num_components	uint16_t	The number of components in the FUP.

Returns

Name (int)	Description
status	0 on success otherwise an error code.

Description

* Initialize a PLDM FWUP manager instance.

2. pldm_fwup_manager_deinit (macro)

```
void pldm_fwup_manager_deinit(struct pldm_fwup_manager *fwup_mgr)
```

Parameters

Name	Type	Description
fwup_mgr	struct pldm_fwup_manager *	The PLDM FWUP manager instance to deinitialize.

Description

* Deinitialize a PLDM FWUP manager instance.

3. reset_get_cmd_state (macro)

```
void reset_get_cmd_state(struct pldm_fwup_protocol_multipart_transfer *get_cmd_state)
```

Parameters

Name	Type	Description
get_cmd_state	struct pldm_fwup_protocol_multipart_transfer *	The state of the multipart transfer.

Description

* Reset a Get command multipart transfer state.