

Put File Pillar Requirements

Requirements on Get File for all pillars that offer data file preservation

Requirements

The pillar must support the mandatory parts of the Put File functionality

- The pillar must understand the **IdentifyPillarsForPutFile** and **PutFile** primitives and be able to reply to these and find and retrieve data from the given file address.
 - must recognise **IdentifyPillarsForPutFileRequest** and **PutFileRequest** messages
 - must be able to send **IdentifyPillarsForPutFileResponse** messages containing **PillarID**
 - must be able to send **PutFileResponse** and **PutFileComplete** messages
 - must be able to download data using the File Exchange protocol

Optional features

All optional features can be set as requirements by the Service Level Agreement

- The **IdentifyPillarsForPutFileReply** message may optionally contain **PillarChecksumType** and **TimeToDeliver** parameters
- The **PutFileRequest** may optionally contain a **ChecksumForCheck** parameter. If requested in the SLA, the pillar should compare this with the checksum calculated on the received file. This should only be used to ensure that the data has been transferred
- The **PutFileRequest** may optionally contain a **SaltForCheck** parameter. If requested in the SLA, the pillar can use this to generate a new checksum when the file reach preservation media. This checksum can be returned in the **PutFileComplete** message.
- The **PutFileResponse** can be used by the pillar to convey non-alarm information back to the client. Examples:
 1. a. This pillar is quite busy. To acknowledge that the **PutFileRequest** is received ok, it may send a **PutFileResponse** with **ResponseCode** "OK" to the client. (And this is an obvious occasion to use the **TimeToDeliver** parameter.!)
 - b. This pillar buffers data for days or weeks before writing them to preservation media. To acknowledge that the file is received ok, and is currently stored in the buffer, it may send a **PutFileResponse** with **ResponseCode** "OK_BF" to the client.
- The **PutFileComplete** can be used to signal the client, that as far as the pillar is concerned, the Put File operation is considered ended.
- Both **PutFileResponse** and **PutFileComplete** messages may be used to inform the client of certain types of error situations. These uses are considered supplementary. The obligatory way of signal error situations is through the alarm subsystem.
- The Put File message is designed to handle **ingest of multiple files**. This is an optional feature.
- The Put File message is designed to handle **paging of a single file into multiple parts**. This is an optional feature.