WAKxxx

Trigger: OSF – 'w' in WK stage

Summary:

This stage will perform "pod–whacking", or segmentation of POD files into one exposure per POD file, for each POD entry in the **.podlist** file.

NOTE: This segmentation is only needed for instruments where a POD file can contain data for more than one exposure (e.g. STIS, NICMOS). It does NOT apply to instruments that have a one–to–one mapping from PODs to exposures (e.g. WFPC–2). For instruments that don't need pod–whacking, this stage does a simple copy of the archive–retrieved POD file to the POD file name required for submission to the science pipeline. A copy is done so that the file ownership is changed from the archive (which placed the files there) to the OTFR pipeline operator.

This process will read the contents of the **.podlist** file, and for every POD entry found it will run the **podwhacker** OPUS utility to extract exposure data into one POD file per exposure. The newly created POD file will be named **exposure_podname.pod**, and will be written to the OTFR request–specific directory. Any EDT entries in the **.podlist** file will be skipped over.

Pod segmentation is needed for several reasons

1. This results in only the data requested by the user being fed into the HST science pipeline. Without pod–whacking, some POD files would contain extra science exposures not requested by the archive user, which would consume extra resources in the science pipeline, and clutter up the OMG with irrelevant entries.
2. This allows unique exposure–pod filenames to be created, which make it more clear which pod files pertain to which exposures. This can help in debugging problem requests.
3. This results in fewer potential collisions in the HST science pipeline when separate OTFR requests need different exposures from the same POD file.

Before running **podwhacker,** any pre–existing file matching the name of the output POD file will be deleted. After **podwhacker** is run, and its return code is checked for error, an additional check is made to be sure that the output POD file does now exist. If either a bad return code is received, or the intended output POD file is not found, the OSF is marked for error.

Success Conditions

A new set of POD files will appear in the OTFR request–specific directory, with names like **exposure_podname.pod**. There should be one for each POD entry in the **.podlist** file.
OSF.WK = c
OSF.MV = w

Error Conditions

Failure to open the **.podlist** file.
Failure to find a POD file listed in the **.podlist** file.
Failure during **podwhacker** execution.
Failure to find resulting **podwhacker** output file.
OSF.WK = e