**Persistent Storage (file\_offload) bplib/stor/store/file\_offload.c**

| File Offload API | Description |
| --- | --- |
| static BPL\_file\_offload\_svc\_t \*bplib\_file\_offload\_instantiate(BPL\_file\_offload\_state\_t parent, void \*init\_arg); | Create a file offload service and return a pointer to the service |
| static int BPL\_file\_offload\_configure(BPL\_file\_offload\_svc\_t \*svc, int key, BPL\_cache\_module\_valtype\_t vt, const void \*val) | Configure a file offload service. The only configuration option is to set the name of the base directory. key must be BPL\_cache\_confkey\_offload\_base\_dir. vt - value type is ignored and assumed to be BPL\_cache\_module\_valtype\_string. val is a null-terminated string containing the base directory name. |
| static int BPL\_file\_offload\_query(BPL\_file\_offload\_svc\_t \*svc, int key, BPL\_cache\_module\_valtype\_t vt, const void \*\*val) | Query and return housekeeping data. |
| static int BPL\_file\_offload\_start(BPL\_file\_offload\_svc\_t \*svc) | Start a file offload service. |
| static int BPL\_file\_offload\_stop(BPL\_file\_offload\_svc\_t \*svc) | Stop a file offload service. |
| static int BPL\_file\_offload\_offload(BPL\_file\_offload\_svc\_t \*svc, bp\_sid\_t \*sid, BPL\_file\_offload\_pblk\_t \*pblk) | Offload a bundle to a file. sid is the Storage ID for the bundle. pblk is a pointer to the bundle Primary Block. |
| static int BPL\_file\_offload\_restore(BPL\_file\_offload\_svc\_t \*svc, bp\_sid\_t sid, BPL\_file\_offload\_pblk\_t \*\*pblk\_out) | Restore a bundle from a file. sid is the Storage ID of the bundle. pblk\_out returns the restored bundle. |
| static int BPL\_file\_offload\_delete(BPL\_file\_offload\_svc\_t \*svc, bp\_sid\_t sid) | Delete a bundle from persistent storage by Storage ID. |

**Based on file\_offload.c bplib/store/file\_offload.c**

static bplib\_mpool\_block\_t \*bplib\_file\_offload\_instantiate(bplib\_mpool\_ref\_t parent, void \*init\_arg);  
static int bplib\_file\_offload\_configure(bplib\_mpool\_block\_t \*svc, int key, bplib\_cache\_module\_valtype\_t vt, const void \*val);  
static int bplib\_file\_offload\_query(bplib\_mpool\_block\_t \*svc, int key, bplib\_cache\_module\_valtype\_t vt, const void \*\*val);  
static int bplib\_file\_offload\_start(bplib\_mpool\_block\_t \*svc);  
static int bplib\_file\_offload\_stop(bplib\_mpool\_block\_t \*svc);  
static int bplib\_file\_offload\_offload(bplib\_mpool\_block\_t \*svc, bp\_sid\_t \*sid, bplib\_mpool\_block\_t \*pblk);  
static int bplib\_file\_offload\_restore(bplib\_mpool\_block\_t \*svc, bp\_sid\_t sid, bplib\_mpool\_block\_t \*\*pblk\_out);  
static int bplib\_file\_offload\_release(bplib\_mpool\_block\_t \*svc, bp\_sid\_t sid);