This document contains a list of all the business process logic (BPL) that is available and accesible for scripting. Please see the appliance archiver [details](https://slacmshankar.github.io/epicsarchiver_docs/details.html#scripting) for a high level description and a small example. All the BPL is typically accessible thru the following process

1. Construct a URL (pointing to the mgmt web app) using the list below. For most BPL, you can point to any appliance in the cluster.
2. Use a HTTP client to get the contents of this URL. For example, in python, the requests package is very well suited for this.
3. The contents of the response are typically JSON and can be parsed using a standard JSON parser. For example, in python you can use the standard json package.

**/getAllPVs**

- Get all the PVs in the cluster. Note this call can return millions of PVs

For more details, please see the [javadoc](https://slacmshankar.github.io/epicsarchiver_docs/api/org/epics/archiverappliance/mgmt/bpl/GetAllPVs.html)

pv

An optional argument that can contain a [GLOB](http://en.wikipedia.org/wiki/Glob_%28programming%29) wildcard. We will return PVs that match this GLOB. For example, if pv=KLYS\*, the server will return all PVs that start with the string KLYS. If both pv and regex are unspecified, we match against all PVs.

regex

An optional argument that can contain a [Java regex](http://docs.oracle.com/javase/7/docs/api/java/util/regex/Pattern.html) wildcard. We will return PVs that match this regex. For example, if pv=KLYS\*, the server will return all PVs that start with the string KLYS.

limit

An optional argument that specifies the number of matched PV's that are retured. If unspecified, we return 500 PV names. To get all the PV names, (potentially in the millions), set limit to –1.

**/getAllExpandedPVNames**

- Get all expanded PV names in the cluster. This is targeted at automation and should return the PV's being archived, the fields, .VAL's, aliases and PV's in the archive workflow. Note this call can return 10's of millions of names.

For more details, please see the [javadoc](https://slacmshankar.github.io/epicsarchiver_docs/api/org/epics/archiverappliance/mgmt/bpl/GetAllExpandedPVNames.html)

**/getPVStatus**

- Get the status of a PV.

For more details, please see the [javadoc](https://slacmshankar.github.io/epicsarchiver_docs/api/org/epics/archiverappliance/mgmt/bpl/GetPVStatusAction.html)

pv

The name(s) of the pv for which status is to be determined. If a pv is not being archived, you should get back a simple JSON object with a status string of "Not being archived." You can also pass in GLOB wildcards here and multiple PVs as a comma separated list. If you have more PVs that can fit in a GET, send the pv's as a CSV pv=pv1,pv2,pv3 as the body of a POST.

**/getPVTypeInfo**

- Get the type info for a given PV. In the archiver appliance terminology, the PVTypeInfo contains the various archiving parameters for a PV.

For more details, please see the [javadoc](https://slacmshankar.github.io/epicsarchiver_docs/api/org/epics/archiverappliance/mgmt/bpl/GetPVTypeInfo.html)

pv

The name of the pv.

**/archivePV**

- Archive one or more PV's.

For more details, please see the [javadoc](https://slacmshankar.github.io/epicsarchiver_docs/api/org/epics/archiverappliance/mgmt/bpl/ArchivePVAction.html)

pv

The name of the pv to be archived. If archiving more than one PV, use a comma separated list. You can also send the PV list as part of the POST body using standard techniques. If you need to specify different archiving parameters for each PV, send the data as a JSON array (remember to send the content type correctly).

samplingperiod

The sampling period to be used. Optional, default value is 1.0 seconds.

samplingmethod

The sampling method to be used. For now, this is one of SCAN or MONITOR. Optional, default value is MONITOR.

controllingPV

The controlling PV for coditional archiving. Optional; if unspecified, we do not use conditional archiving.

policy

Override the policy execution process and use this policy instead. Optional; if unspecified, we go thru the normal policy execution process.

appliance

Optional; you can specify an appliance in a cluster. If specified (value is the identity of the appliance), the sampling and archiving are done on the specified appliance.

**/pauseArchivingPV**

- Pause archiving the specified PV. This also tears down the CA channel for this PV.

For more details, please see the [javadoc](https://slacmshankar.github.io/epicsarchiver_docs/api/org/epics/archiverappliance/mgmt/bpl/PauseArchivingPV.html)

pv

The name of the pv. You can also pass in GLOB wildcards here and multiple PVs as a comma separated list. If you have more PVs that can fit in a GET, send the pv's as a CSV pv=pv1,pv2,pv3 as the body of a POST.

**/resumeArchivingPV**

- Resume archiving the specified PV.

For more details, please see the [javadoc](https://slacmshankar.github.io/epicsarchiver_docs/api/org/epics/archiverappliance/mgmt/bpl/ResumeArchivingPV.html)

pv

The name of the pv. You can also pass in GLOB wildcards here and multiple PVs as a comma separated list. If you have more PVs that can fit in a GET, send the pv's as a CSV pv=pv1,pv2,pv3 as the body of a POST.

**/getStoresForPV**

- Gets the names and definitions of the data stores for this PV. Every store in a PV's typeinfo is expected to have a name - this is typically "name=STS" or something similar. This call returns the names of all the stores for a PV with their URI representations as a dictionary.

For more details, please see the [javadoc](https://slacmshankar.github.io/epicsarchiver_docs/api/org/epics/archiverappliance/mgmt/bpl/GetStoresForPV.html)

pv

The name of the pv.

**/modifyStoreURLForPV**

- Changes the store for this particular PV. Note this only changes the PVTypeInfo; it does not change any data/files so one could lose data using this call.

For more details, please see the [javadoc](https://slacmshankar.github.io/epicsarchiver_docs/api/org/epics/archiverappliance/mgmt/bpl/ModifyStoreURLForPV.html)

pv

The name of the pv

storage

The name of the store to change ( for example, MTS )

plugin\_url

The new URL specification for this store; this is what you would have used in the policy file and is something that can be understood by [StoragePluginURLParser](https://slacmshankar.github.io/epicsarchiver_docs/api/org/epics/archiverappliance/config/StoragePluginURLParser.html).

**/consolidateDataForPV**

- Consolidate the data for this PV until the specified store. The PV needs to be paused first.

For more details, please see the [javadoc](https://slacmshankar.github.io/epicsarchiver_docs/api/org/epics/archiverappliance/mgmt/bpl/ConsolidatePBFilesForOnePV.html)

pv

The name of the pv.

storage

The name of the store until which we'll consolidate data. This is typically a string like STS or MTS. To get a list of names of stores for a PV, please see /getStoresForPV

**/deletePV**

- Stop archiving the specified PV. The PV needs to be paused first.

For more details, please see the [javadoc](https://slacmshankar.github.io/epicsarchiver_docs/api/org/epics/archiverappliance/mgmt/bpl/DeletePV.html)

pv

The name of the pv.

deleteData

Should we delete the data that has already been recorded. Optional, by default, we do not delete the data for this PV. Can be true or false.

**/abortArchivingPV**

- Abort any pending requests for archiving this PV.

For more details, please see the [javadoc](https://slacmshankar.github.io/epicsarchiver_docs/api/org/epics/archiverappliance/mgmt/bpl/AbortArchiveRequest.html)

pv

The name of the PV.

**/changeArchivalParameters**

- Change the archival parameters for a PV.

For more details, please see the [javadoc](https://slacmshankar.github.io/epicsarchiver_docs/api/org/epics/archiverappliance/mgmt/bpl/ChangeArchivalParamsAction.html)

pv

The name of the pv.

samplingperiod

The new sampling period in seconds.

samplingmethod

The new sampling method For now, this is one of SCAN or MONITOR.

**/getPVDetails**

- Get a lot of detailed statistics for a PV. The returned JSON is very UI friendly; but should be usable in a scripting environment.

For more details, please see the [javadoc](https://slacmshankar.github.io/epicsarchiver_docs/api/org/epics/archiverappliance/mgmt/bpl/reports/PVDetails.html)

pv

The name of the pv.

**/getApplianceInfo**

- Get the appliance information for the specified appliance.

For more details, please see the [javadoc](https://slacmshankar.github.io/epicsarchiver_docs/api/org/epics/archiverappliance/mgmt/bpl/GetApplianceInfo.html)

id

The identity of the appliance for which we are requesting information. This is the same string as the identity element in the appliances.xml that identifies this appliance.

**/getAppliancesInCluster**

- Get the appliance information for all the appliances in the cluster that are active.

For more details, please see the [javadoc](https://slacmshankar.github.io/epicsarchiver_docs/api/org/epics/archiverappliance/mgmt/bpl/GetAppliancesInCluster.html)

**/renamePV**

- Rename this pv to a new name. The PV needs to be paused first.

For more details, please see the [javadoc](https://slacmshankar.github.io/epicsarchiver_docs/api/org/epics/archiverappliance/mgmt/bpl/RenamePVAction.html)

pv

The current name of the PV.

newname

The new name of the PV.

**/reshardPV**

- This BPL reassigns the PV to another appliance. This is a complex BPL that also moves data over to the other appliance. For more details on the sequence of steps, see the Javadoc for the BPL.

For more details, please see the [javadoc](https://slacmshankar.github.io/epicsarchiver_docs/api/org/epics/archiverappliance/mgmt/bpl/ReshardPV.html)

pv

The name of the pv. The PV needs to be paused first and will remain in a paused state after the resharding is complete.

appliance

The new appliance to assign the PV to. This is the same string as the identity element in the appliances.xml that identifies this appliance.

storage

The name of the store until which we'll consolidate data before resharding. The data is moved over to the store with the same name on the new appliance. This is typically a string like LTS.

**/appendAndAliasPV**

- This BPL appends the data for an older PV into a newer PV. The older PV is deleted and an alias mapping the older PV name to the new PV is added.

For more details, please see the [javadoc](https://slacmshankar.github.io/epicsarchiver_docs/api/org/epics/archiverappliance/mgmt/bpl/AppendAndAliasPV.html)

olderpv

The name of the older pv. The data for this PV will be appended to the newer PV and then deleted.

newerpv

The name of the newer pv.

storage

The name of the store until which we'll consolidate data before appending. This is typically a string like LTS.

**/addAlias**

- Add an alias for the specified PV.

For more details, please see the [javadoc](https://slacmshankar.github.io/epicsarchiver_docs/api/org/epics/archiverappliance/mgmt/bpl/AddAliasAction.html)

pv

The real name of the pv.

aliasname

The alias name of the pv. Note, we cannot have a PVTypeInfo mapped to the alias name.

**/removeAlias**

- Remove an alias for the specified PV. This is only supported for PVs who have completed their archive PV workflow.

For more details, please see the [javadoc](https://slacmshankar.github.io/epicsarchiver_docs/api/org/epics/archiverappliance/mgmt/bpl/RemoveAliasAction.html)

pv

The real name of the pv.

aliasname

The alias name of the pv.

**/getAllAliases**

- Get all the aliases in the cluster and the PV's they are mapped to.

For more details, please see the [javadoc](https://slacmshankar.github.io/epicsarchiver_docs/api/org/epics/archiverappliance/mgmt/bpl/GetAllAliasesAction.html)

**/skipAliasCheck**

- For PVs that are still in the archive PV workflow, skip the alias check where we examine the .NAME field to determine the real name. This is useful if you have pCAS servers that overload the .NAME field for something else.

For more details, please see the [javadoc](https://slacmshankar.github.io/epicsarchiver_docs/api/org/epics/archiverappliance/mgmt/bpl/SkipAliasCheckAction.html)

pv

The name of the pv as it is in the archive PV workflow.

**/changeTypeForPV**

- Change the type of the PV to the specified type. The PV needs to be paused first. For best results, consolidate all the data to one store. Note, this is actually changing the data so you should make a backup just in case. There is every chance that this may leave the data for this PV in an inconsistent state. It is also possible that the plugin may do nothing in which case you may have to rename the existing PV to a new PV; delete this PV and issue a fresh archival request.

For more details, please see the [javadoc](https://slacmshankar.github.io/epicsarchiver_docs/api/org/epics/archiverappliance/mgmt/bpl/ChangeTypeForPV.html)

pv

The name of the pv.

newtype

The new type one of the ArchDBRTypes. For example, DBR\_SCALAR\_DOUBLE.

**/mergeInData**

- Manually merges in data from an external appliance that is used for failover.

For more details, please see the [javadoc](https://slacmshankar.github.io/epicsarchiver_docs/api/org/epics/archiverappliance/mgmt/bpl/MergeInDataFromExternalStore.html)

pv

The name of the pv; the PV needs to be paused. It's probably also a good idea to consolidate the data to the store being used for merging.

other

This is the data\_retrieval\_url of the other appliance.

storage

The name of the data store (LTS etc) to merge the data into. Please consolidate the data into this store to make sure merging in later data from the remote store does not result in losing data in the other stores on this appliance.

from

The start time for the retrieval request to the other server defaults to 2 months ago.

to

The end time for the retrieval request to the other server defaults to a month from now.

**/resetFailoverCaches**

- Reset the failover caches for all the retrieval components in this cluster.

For more details, please see the [javadoc](https://slacmshankar.github.io/epicsarchiver_docs/api/org/epics/archiverappliance/mgmt/bpl/ResetFailoverCaches.html)

**/getVersions**

- Get the versions of the various components for this appliance.

For more details, please see the [javadoc](https://slacmshankar.github.io/epicsarchiver_docs/api/org/epics/archiverappliance/mgmt/bpl/GetVersions.html)

id

The identity of the appliance for which we are requesting information. This is the same string as the identity element in the appliances.xml that identifies this appliance.

**/modifyMetaFields**

- Modify the fields (HIHI, LOLO etc) being archived as part of the PV. PV needs to be paused first.

For more details, please see the [javadoc](https://slacmshankar.github.io/epicsarchiver_docs/api/org/epics/archiverappliance/mgmt/bpl/ModifyMetaFieldsAction.html)

pv

The real name of the pv.

command

A command is a verb followed by a list of fields, all of them comma separated. Possible verbs are add, remove and clear. For example add,ADEL,MDEL will add the fields ADEL and MDEL if they are not already present. clear clears all the fields. You can have any number of commands.

**/getNamedFlag**

- Get the value of the named flag specified by name

For more details, please see the [javadoc](https://slacmshankar.github.io/epicsarchiver_docs/api/org/epics/archiverappliance/mgmt/bpl/NamedFlagsGet.html)

name

the name of the named flag.

**/setNamedFlag**

- Set the value of the named flag specified by name

For more details, please see the [javadoc](https://slacmshankar.github.io/epicsarchiver_docs/api/org/epics/archiverappliance/mgmt/bpl/NamedFlagsSet.html)

name

the name of the named flag.

value

Either true of false; something that Boolean.parse can understand.

**/getTypeInfoKeys**

- Get a list of all the PV names that have a PVTypeInfo.

For more details, please see the [javadoc](https://slacmshankar.github.io/epicsarchiver_docs/api/org/epics/archiverappliance/mgmt/bpl/GetPVTypeInfoKeys.html)

**/getNeverConnectedPVs**

- Get a list of PVs that have never connected.

For more details, please see the [javadoc](https://slacmshankar.github.io/epicsarchiver_docs/api/org/epics/archiverappliance/mgmt/bpl/reports/NeverConnectedPVsAction.html)

**/getMetaGets**

- Get a list of PVs that are currently in METAINFO\_REQUESTED state.

For more details, please see the [javadoc](https://slacmshankar.github.io/epicsarchiver_docs/api/org/epics/archiverappliance/mgmt/bpl/reports/MetaGetsAction.html)

**/getCurrentlyDisconnectedPVs**

- Get a list of PVs that are currently disconnected.

For more details, please see the [javadoc](https://slacmshankar.github.io/epicsarchiver_docs/api/org/epics/archiverappliance/mgmt/bpl/reports/CurrentlyDisconnectedPVs.html)

**/getEventRateReport**

- Return a list of PVs sorted by descending event rate.

For more details, please see the [javadoc](https://slacmshankar.github.io/epicsarchiver_docs/api/org/epics/archiverappliance/mgmt/bpl/reports/EventRateReport.html)

limit

Limit this report to this many PVs per appliance in the cluster. Optional, if unspecified, there are no limits enforced.

**/getStorageRateReport**

- Return a list of PVs sorted by descending storage rate.

For more details, please see the [javadoc](https://slacmshankar.github.io/epicsarchiver_docs/api/org/epics/archiverappliance/mgmt/bpl/reports/StorageRateReport.html)

limit

Limit this report to this many PVs per appliance in the cluster. Optional, if unspecified, there are no limits enforced.

**/getRecentlyAddedPVs**

- Return a list of PVs sorted by descending PVTypeInfo creation timestamp.

For more details, please see the [javadoc](https://slacmshankar.github.io/epicsarchiver_docs/api/org/epics/archiverappliance/mgmt/bpl/reports/RecentlyAddedPVs.html)

limit

Limit this report to this many PVs per appliance in the cluster. Optional, if unspecified, there are no limits enforced.

**/getRecentlyModifiedPVs**

- Return a list of PVs sorted by descending PVTypeInfo modification timestamp.

For more details, please see the [javadoc](https://slacmshankar.github.io/epicsarchiver_docs/api/org/epics/archiverappliance/mgmt/bpl/reports/RecentlyChangedPVs.html)

limit

Limit this report to this many PVs per appliance in the cluster. Optional, if unspecified, there are no limits enforced.

**/getLostConnectionsReport**

- Return a list of PVs sorted by number of times we've lost/reestablished connections (descending) to the IOC hosting the PV. Can be used to determine if the IOC is being overloaded.

For more details, please see the [javadoc](https://slacmshankar.github.io/epicsarchiver_docs/api/org/epics/archiverappliance/mgmt/bpl/reports/LostConnectionsReport.html)

limit

Limit this report to this many PVs per appliance in the cluster. Optional, if unspecified, there are no limits enforced.

**/getSilentPVsReport**

- Return a list of PVs sorted by the timestamp of the last event received (descending).

For more details, please see the [javadoc](https://slacmshankar.github.io/epicsarchiver_docs/api/org/epics/archiverappliance/mgmt/bpl/reports/SilentPVReport.html)

limit

Limit this report to this many PVs per appliance in the cluster. Optional, if unspecified, there are no limits enforced.

**/getPVsByDroppedEventsTimestamp**

- Return a list of PVs sorted by number of times we've lost events because of incorrect timestamps.

For more details, please see the [javadoc](https://slacmshankar.github.io/epicsarchiver_docs/api/org/epics/archiverappliance/mgmt/bpl/reports/DroppedEventsTimestampReport.html)

limit

Limit this report to this many PVs per appliance in the cluster. Optional, if unspecified, there are no limits enforced.

**/getPVsByDroppedEventsBuffer**

- Return a list of PVs sorted by number of times we've lost events because of buffer overflows; perhaps from a mismatch between the event rate and the sampling rate (as determined from the sampling period).

For more details, please see the [javadoc](https://slacmshankar.github.io/epicsarchiver_docs/api/org/epics/archiverappliance/mgmt/bpl/reports/DroppedEventsBufferOverflowReport.html)

limit

Limit this report to this many PVs per appliance in the cluster. Optional, if unspecified, there are no limits enforced.

**/getPVsByDroppedEventsTypeChange**

- Return a list of PVs sorted by number of times we've lost events because of changes in type of the underlying PV. This does not include PVs that have been paused.

For more details, please see the [javadoc](https://slacmshankar.github.io/epicsarchiver_docs/api/org/epics/archiverappliance/mgmt/bpl/reports/DroppedEventsTypeChangeReport.html)

limit

Limit this report to this many PVs per appliance in the cluster. Optional, if unspecified, there are no limits enforced.

**/getSlowChangingPVsWithDroppedEvents**

- Return a list of PVs which have lost more events than events from the IOC sorted by number of times we've lost events. This does not include PVs that have been paused.

For more details, please see the [javadoc](https://slacmshankar.github.io/epicsarchiver_docs/api/org/epics/archiverappliance/mgmt/bpl/reports/SlowChangingPVsWithDroppedEvents.html)

limit

Limit this report to this many PVs per appliance in the cluster. Optional, if unspecified, there are no limits enforced.

**/getPausedPVsReport**

- Return a list of PVs that are currently paused.

For more details, please see the [javadoc](https://slacmshankar.github.io/epicsarchiver_docs/api/org/epics/archiverappliance/mgmt/bpl/reports/PausedPVsReport.html)

limit

Limit this report to this many PVs per appliance in the cluster. Optional, if unspecified, there are no limits enforced.

**/getArchivedWaveforms**

- Get a list of waveform PVs that are currently being archived.

For more details, please see the [javadoc](https://slacmshankar.github.io/epicsarchiver_docs/api/org/epics/archiverappliance/mgmt/bpl/reports/WaveformPVsAction.html)

**/getTimeSpanReport**

- Archiving time span; when we first added the PV to the system, last known timestamp (if available), paused or not.

For more details, please see the [javadoc](https://slacmshankar.github.io/epicsarchiver_docs/api/org/epics/archiverappliance/mgmt/bpl/reports/TimeSpanReport.html)

**/getMatchingPVsForThisAppliance**

- Get matching PV's for this appliance. Specify one of pv or regex. If both are specified, we only apply the pv wildcard. If neither is specified, we return an empty list.

For more details, please see the [javadoc](https://slacmshankar.github.io/epicsarchiver_docs/api/org/epics/archiverappliance/mgmt/bpl/GetMatchingPVsForAppliance.html)

pv

An optional argument that can contain a [GLOB](http://en.wikipedia.org/wiki/Glob_%28programming%29) wildcard. We will return PVs that match this GLOB. For example, if pv=KLYS\*, the server will return all PVs that start with the string KLYS.

regex

An optional argument that can contain a [Java regex](http://docs.oracle.com/javase/7/docs/api/java/util/regex/Pattern.html) wildcard. We will return PVs that match this regex. For example, if pv=KLYS.\*, the server will return all PVs that start with the string KLYS.

limit

An optional argument that specifies the number of matched PV's that are returned. If unspecified, we return 500 PV names. To get all the PV names, (potentially in the millions), set limit to 1.

**/putPVTypeInfo**

- Updates the typeinfo for the specified PV. Note this merely updates the typeInfo. It does not have any logic to react to changes in the typeinfo. That is, don't assume that the PV is automatically paused just because you changed the isPaused to true. This is meant to be used in conjuction with other BPL to implement site-specific BPL in external code (for example, python). This can also be used to add PVTypeInfo's into the system; support for this is experimental. The new PVTypeInfo's are automatically paused before adding into the system. Logically, you have to specify at least one of override or createnew.

For more details, please see the [javadoc](https://slacmshankar.github.io/epicsarchiver_docs/api/org/epics/archiverappliance/mgmt/bpl/PutPVTypeInfo.html)

pv

The name of the pv.

override

If the PVTypeInfo for this PV already exists, do you want to update it or return an error? By default, this is false.

createnew

If the PVTypeInfo for this PV does not exist, do you want to create a new one or return an error? By default, this is false.

**/unarchivedPVs**

- Given a list of PVs, determine those that are not being archived/have pending requests/have aliases.

For more details, please see the [javadoc](https://slacmshankar.github.io/epicsarchiver_docs/api/org/epics/archiverappliance/mgmt/bpl/UnarchivedPVsAction.html)

pv

A list of pv names. Send as a CSV using a POST or JSON array.

**/archivedPVs**

- Given a list of PVs, determine those that are being archived.

For more details, please see the [javadoc](https://slacmshankar.github.io/epicsarchiver_docs/api/org/epics/archiverappliance/mgmt/bpl/ArchivedPVsAction.html)

pv

A list of pv names. Send as a CSV using a POST.

**/archivedPVsNotInList**

- Given a list of PVs, determine those that are being archived but are not in the incoming list.

For more details, please see the [javadoc](https://slacmshankar.github.io/epicsarchiver_docs/api/org/epics/archiverappliance/mgmt/bpl/ArchivedPVsNotInListAction.html)

pv

A list of pv names. Send as a CSV using a POST, or as a JSON.