

APPSYNC WITH EPIC CACHÉ

ABSTRACT

This white paper discusses how EMC AppSync integrates with Epic Caché databases on EMC primary storage arrays. The objective solution provides for an on-going availability of critical user data in an agile environment.

March, 2017

The information in this publication is provided “as is.” Dell Inc. makes no representations or warranties of any kind with respect to the information in this publication, and specifically disclaims implied warranties of merchantability or fitness for a particular purpose.

Use, copying, and distribution of any software described in this publication requires an applicable software license.

Copyright © 2016 Dell Inc. or its subsidiaries. All Rights Reserved. Dell, EMC, and other trademarks are trademarks of Dell Inc. or its subsidiaries. Other trademarks may be the property of their respective owners. Published in the USA. February, 2017 White Paper, H15954.1

EMC believes the information in this document is accurate as of its publication date. The information is subject to change without notice.

TABLE OF CONTENTS

ABSTRACT	1
EXECUTIVE SUMMARY	4
Audience	4
INTRODUCTION	4
APPSYNC OVERVIEW	4
Terms and Definitions	4
AppSync Operations	5
APPSYNC AND EPIC DATABASES	5
CONFIGURING APPSYNC FOR EPIC CACHÉ	6
Understanding Object Layout.....	6
AppSync's Application Support	6
The Configuration.....	6
APPSYNC WORKFLOW PHASES	7
APPSYNC CALLOUTS	8
Configuration Details	9
Callout Example	9
Unmount Callout.....	10
CREATE CACHÉ COPIES	10
Service Plans	10
FileSystems Service Plan for Epic Caché.....	10
Configuring the Filesystem Service Plan	11
Running Service Plans.....	11
Repurposing Workflows	12
Repurposing Mount.....	13
Repurposing Refresh	13
ON-DEMAND MOUNTING	13
POST-MOUNT SCRIPT	15
VPLEX INTEGRATION	15

EXECUTIVE SUMMARY

This document provides technical information about integrating Epic Caché with EMC AppSync software and EMC primary storage systems. This document also provides guidance while deploying the AppSync software in environments where application data is protected across EMC RecoverPoint and EMC VPLEX distributed environments. AppSync offers an exceptionally efficient and beneficial integrated copy management solution, for the most critical electronic health record (EHR) systems, such as Epic Caché.

AUDIENCE

This white paper is intended for both storage administrators as well as Epic Caché end users who need to setup and/or manage AppSync's copy management solutions provisioned on EMC primary storage.

INTRODUCTION

EMC AppSync is software that enables Integrated Copy Data Management (iCDM) with EMC's primary storage systems. AppSync simplifies and automates the process of generating and consuming copies of production data. By abstracting the underlying storage and replication technologies, and through deep application integration, AppSync empowers application owners to satisfy copy demand for operational recovery and data repurposing. In turn, storage administrators need only be concerned with initial setup and policy management, resulting in an agile, frictionless environment. AppSync automatically discovers application environments and structures, maps them through the virtualization layer, when applicable, to the underlying storage devices. AppSync then orchestrates all the activities of copy management, from copy creation to user validation through mounting at the target host and launching scripts to recover the application, if desired. Supported workflows also include refresh, expire, and restore operations.

APPSYNC OVERVIEW

This overview provides general information about the various AppSync components and their roles as a basis for more specific discussions. Additional clarifications will follow, or for more detailed AppSync related information, refer to the AppSync product and user guides found on support.emc.com. AppSync is composed of three main components:

- **AppSync Server** - Stores all information about users, hosts, storage systems, copies, and ongoing operations in a built-in PostgreSQL repository. The server component orchestrates copies through service plans and repurposing workflows. The AppSync server can be installed on a physical or virtual host, and interacts directly with the storage and virtualization components.
- **AppSync User Interface (UI) or Console** – The user interface allowing end-users to interact with, and manage, the product. These components include an internet browser based interface (most common), command line interface (CLI), REST API support, and VSI integration.
- **AppSync Agent Plug-In** — Installed on all source and mount hosts, which interacts with the host operating system and application layers to create, mount, recover, or restore copies of mission-critical data.

There are several AppSync components which deal with specific aspects of the environment. The server-to-storage services component and the various application agent plug-ins are the primary focus of this white paper.

TERMS AND DEFINITIONS

AppSync uses industry terms, which may have a slightly different meaning, terms which have a very specific purpose, versus general meaning. The following terms, defined below, should help provide guidance.

- **Access Control List** – Also known simply as ACLs. The rules, or controls, setup to limit user's access to certain objects.
- **Callout Script** – The scripts initiated during AppSync's create copy phase, allowing customized processes to be executed alongside of AppSync's normal job process – directly integrated into the copy job phase.
- **Expire** – A process flow of removing copies from within the AppSync inventory, also removing the copy on the storage array.
- **Mount Host** – The host where copies are presented. This can be an alternate host or the same host as the source volumes.

- **Mount Point** – A location used by the mount operation, to which uses an existing mounted file system to attach, as a directory tree, the copied volume, appearing simply as a new directory to an existing file system structure. This is the default AppSync mount location which can be changed under the mount options configuration setting.
- **Object** – Any database, file system, application, or datastore, to which AppSync manages. ACLs are applied to these items.
- **Repurposing Workflow** – A copy management workflow process, similar to the Service Plan workflow, providing a multi-generation copy process, both based on native array and RecoverPoint technologies. It is the only way to create an array based copy off of a RecoverPoint image. Repurposing workflows are managed through Copy Management, and are centric to one application at a time.
- **Recover** – A process where a copy is mounted and started, providing a working environment similar to the production, or source, environment. This is not the same as a restore operation, as the recovery occurs with the copy, to an alternate location, rather than overwriting the original copy, or source, volume.
- **Restore** – A process flow of overwriting the source volume with the contents of a copy created previously. This would occur if the source data became corrupt, for example.
- **Service Plan** – A copy management workflow process, and template, for protecting applications. Service Plans are managed in their own menu, and generally support many applications sharing the same settings.
- **Subscribe** – The act performed on/against an object, as to how it gets associated to a service plan – runs with the service plan.

APPSYNC OPERATIONS

AppSync utilizes an agent plug-in component to discover the application layer, mapping it back to the underlying storage infrastructure, also initiating host based commands to ensure consistency, and logical volume management operations. Managing Epic Caché requires the interaction with operating system file systems, and volume management, both at the logical and physical layers. When virtualized or clustered, AppSync requires additional management interactions. Some of the operations AppSync provides are as follows:

- Discovering the file system objects to which copy management tasks are to be acted upon
- Mapping file system objects through the various layers to underlying storage
- Managing storage copy transactions by utilizing scheduled, or on-demand, service plans and repurposing workflows – utilizing the underlying storage technology
- Managing all host based operations, such as scanning, importing, exporting, mounting, unmounting, and running file system validation checks
- Managing storage and virtualization layer device provisioning

APPSYNC AND EPIC DATABASES

AppSync version 3.1, released December 2016, included enhancements targeted at the Epic Caché operational databases by utilizing AppSync's File System Service Plans and Repurposing Workflows. AppSync can also be utilized to support Cogito Clarity and Caboodle analytical databases when utilizing either Microsoft SQL or Oracle, providing the most robust environment support within one pane of glass. AppSync has built-in APIs for both Oracle and SQL, though is not addressed in this white paper. More information about supporting such environments can be found on support.emc.com.

AppSync meets the most critical Epic Caché backup and environment refresh challenges by providing efficient, reliable, and proven copies whenever needed. Backups are able to be automated, and AppSync supports the most agile environments – maintaining the ability to recover quickly from data loss or corruption, in addition to creating additional copies for testing and development. AppSync offers an efficient tool for application owners, potentially alleviating the need for storage administration requests, which can often increase access times.

Configuring AppSync for Epic Caché

UNDERSTANDING OBJECT LAYOUT

In order for AppSync to support Epic Caché environments, it is imperative to understand the underlying application environment. This is in some ways a contrast to supporting SQL or Oracle environments, for example, in the sense that AppSync maps the “application” to the underlying file systems, where supporting Epic Caché requires mapping the file systems directly. Knowing which file systems make up the Epic Caché environment, therefore, is essential, where it is not necessary when managing SQL or Oracle, as one simply needs to know only the particular database name.

In addition, when changing an SQL or Oracle underlying environment, AppSync understands the changes based on the application mapping phase – the application tells AppSync what it needs to replicate, where supporting Epic Caché requires manual re-configuration within AppSync - AppSync will not know about such file systems being added/removed. **Supporting Epic Caché requires knowing exactly which file systems need to be protected**, so if that file system changes, so would the workflow configuration.

The following rules must be acknowledged before configuring Epic Caché workflows with AppSync:

- All devices in a volume group must be copied together – multiple volume groups can be managed together
- All devices in a volume group that are copied must be mounted together
- Restoring copies are restored at the volume group level
- When managing environments with RecoverPoint, consistency is maintained at the RecoverPoint Consistency Group level, so all devices are replicated/restored at the RP CG level – please ensure RecoverPoint best practices are also followed

APPSYNC’S APPLICATION SUPPORT

AppSync provides five different types of application support, Microsoft Exchange and SQL, Oracle, VMware datastores, and file systems (Windows, AIX, and Linux). Since there are no built-in APIs for Epic Caché, all non-native application integration is supported by the use of the File System Service Plans and/or Repurposing workflows. It is with the File System service plans and workflows that application owners can incorporate scripts which freeze, suspend, stop, shutdown, thaw, resume, restart, etc., to provide application layer consistency. Without the utilization of scripts, with the file system plans and workflows, copies would only be considered crash consistent and typically not meet the required consistency level needed for proper recovery.

THE CONFIGURATION

Detailed installation and basic configuration of AppSync can be referenced by reviewing the *AppSync Installation and Configuration Guide*, which is a precursor to this white paper. Understanding how to install AppSync server software, adding storage and licensing, configuring VMware (if applicable) and basic UI management is not discussed in detail, within this white paper. This white paper assumes the basic setup is complete, and will begin by adding the appropriate hosts and configuring the Epic Caché environment.

Assuming the AppSync server is installed and the storage has been configured, supporting the Epic Caché environment requires the following high level configuration, including a few simple rules to understand:

1. Add the servers and pushing the agent plug-ins to all nodes of the environment
 - All source and mount hosts, including all the nodes of a PowerHA cluster
2. Register the service label IP/Name - when managing PowerHA environments only
 - All nodes must first be added
 - Only one service label IP/Name per resource groups should be registered if multiple are utilized in the environment
 - Create copies either through the use of service plans or repurposing workflows – described in more detail in the [Create Caché Copies](#) section below

APPSYNC WORKFLOW PHASES

AppSync provides a simple management tool for copy management on EMC primary storage. The following provides an overview of each of the general phases during the file system service plan and repurposing workflow copy processes. In order to ensure Epic Caché environments are properly copied, please note that in the copy phase, callout scripts are utilized. The callout scripts are explained in more detail, in the following section titled, [AppSync Callouts](#).

Workflows support both physical and/or virtual environments, integrating with the application layer, and coordinating storage operations in order to provide a means for the end user to ensure the copy is valid.

- **Application discovery phase** – This phase always runs, and is not configurable. It examines the source/production server, looking for changes, and ensures the application is available.
- **Application mapping phase** – This phase always runs, and is not configurable. It maps applications to storage/protection services, such as mapping the file systems to volumes on the backend storage array.
- **Pre-copy script phase** – This phase starts before the Create copy phase, and is optionally configurable within the UI. The timing of this script initiation is not conducive to supporting application consistent copies, such as freezing the Epic Caché environment. Scripts at this phase may be non-application related steps, such as when performing preparatory steps before creating a copy.
- **Create copy phase** - Starts when the Pre-copy script phase completes, and always runs. This phase includes a number of configurable settings, such as the number of copies to keep, and storage preferences. Please see the *AppSync User and Administration Guide* for more details on these settings. This phase is broken into three sub-phases, as depicted below.
 - **Freeze callout** – Starts before the array creates a consistent copy of the file system service plan or repurposing workflow only. The freeze callout executes by looking for a file, in a specific location, and if present, runs it. This is discussed in more detail, in the section titled [AppSync Callouts](#), and is core to maintaining a consistent copy of Epic Caché – use with the **instfreeze** scripts supplied by Epic.
 - **Create local copy** - Initiates the creation of snaps or clones on the storage array, or bookmarks when managed with RecoverPoint. This phase always runs.
 - **Thaw callout** – Starts after array creates the consistent copy of the file system, thus allowing the application to resume normal operations. The thaw callout is executed only if present. This is the second aspect to the callout scripts, discussed in more detail, in the [AppSync Callout](#) section, and is core to maintaining a consistent copy of Epic Caché – use with the **instthaw** scripts supplied by Epic.
- **Post-copy script phase** - Starts when the Create copy phase completes and is optional. This is also not generally used for application consistent configurations, as mentioned previously.
- **Unmount callout phase** – This phase only starts if the **Unmount previous copy** phase is configured, and only executes if the appropriate file is present on the mount host, discussed in more detail in the section titled [AppSync Callouts](#). This phase, like all other callout scripts, is only supported with file system service plans and repurposing workflows, as of the time of publication.
- **Unmount previous copy phase** – This phase is optional, and starts after the **Unmount callout phase** completes. This phase unmounts copies only if they were also mounted previously with the same service plan. Any copies manually mounted, are unaffected by this phase.
- **Pre-mount script phase** – This phase is optional, and starts before the Mount copy phase begins.
- **Mount copy phase** – This phase is optional, and starts after the Pre-mount script phase completes.
- **Post-mount script phase** – This phase is optional and starts after the Mount copy phase completes. This phase is typically used to initiate a 3rd party backup application which transfers the copy to another type of long term storage.
- **Unmount callout phase** – This phase only starts if the **Unmount copy phase** (below) is configured, and only executes if the appropriate file is present on the mount host, discussed in more detail in the section titled [AppSync Callouts](#). This phase, like all other callout scripts, is only supported with file system service plans and repurposing workflows, as of the time of publication.
- **Unmount copy phase** - Starts after the Unmount callout phase completes, is optional, and used to limit how long a copy is mounted. This phase is used in conjunction, typically, when a backup operation completes during the post-mount scripting phase.

APPSYNC CALLOUTS

During the Create copy phase, which always runs, AppSync has the ability to incorporate an additional level of control over the application being copied. AppSync offers pre-copy and post-copy scripting options, as seen in the UI, also configurable within the CLI and REST API interfaces, however, these scripting options are not conducive to controlling application freezing and thawing. The differences between **pre/post-copy** scripts, and **callout** scripts, are as follows:

- **Configuration/Initiation**

- Pre/Post-Copy scripts are configured within the AppSync UI, as well as the CLI and REST interfaces.
- Callout scripts are initiated by naming files very specifically and placing them in a very specific location on the production, and/or mount host(s).

- **Application/Use Cases**

- Pre/Post-Copy scripts are fitting for all application types, other than datastores, for pre or post copy processing - performing steps before/after creating a copy, where timing between the two is not a critical consideration.
- Callout scripts are strictly used with the file system Service Plans and Repurposing Workflows - ideal for quiescing (suspending I/O) a database or application that AppSync does not support natively, such as Epic Caché. Use this option when timing between the freeze and thaw needs to be precise, as applies to freezing and thawing an application - suspending I/O to an application should be done with the most minimal amount of time as possible.

- **Occurrence**

- Pre/Post-Copy scripts occur before/after the copy process, so the amount of time between is unknown – dependent upon a number of factors.
- Callout scripts occur during, or within rather, the copy and/or unmount phases (see **Figure 1**), so are appropriate for 3rd party database integration, with the most minimal time lag.

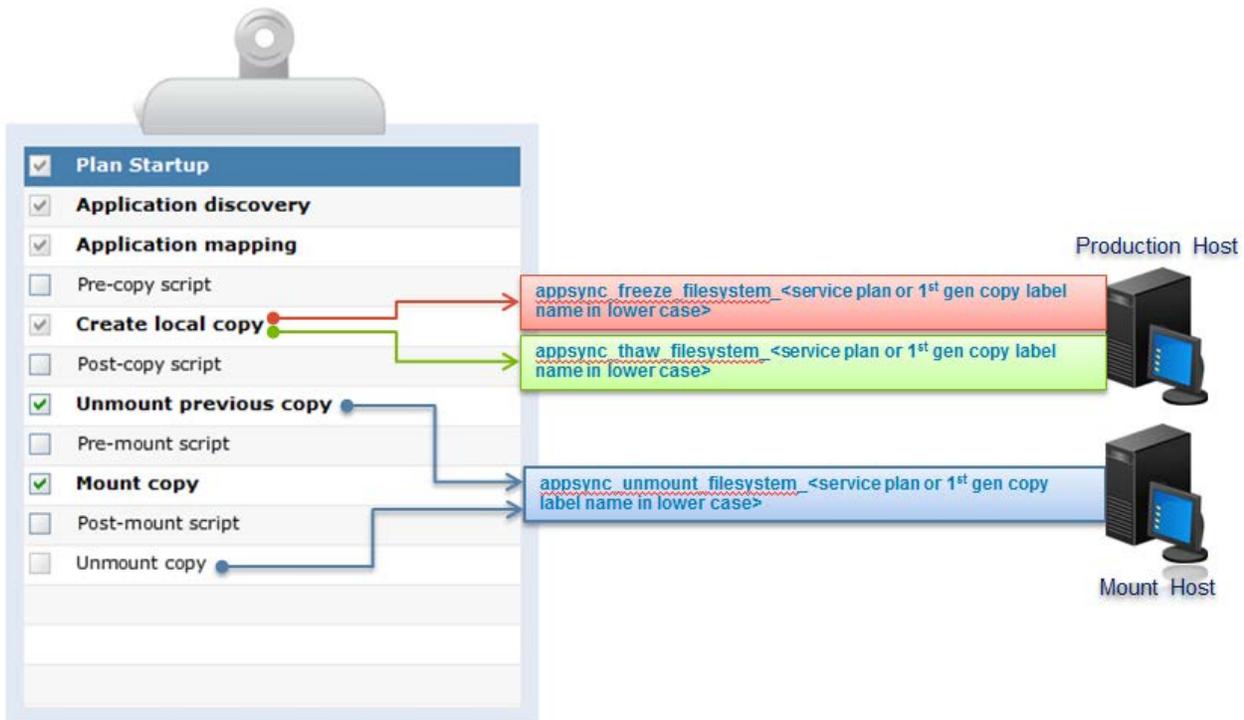


Figure 1 - Callout Script Job Logic

CONFIGURATION DETAILS

Callouts are **specifically named files** which sit on the production and/or mount hosts, in a **specific location**, which are **automatically** called during the **file system copy process**. Callouts are used to initiate customized scripts, to be executed during the copy process, or before unmounting operations, i.e. freezing and thawing the Epic Caché application, and shutting down the application on the mount host, before refreshing the copy. AppSync supports the execution of customized scripts, however, cannot support what those particular scripts perform. Always review the *AppSync User and Administration Guide* for more details.

When configuring callout scripts for supporting the creation of Epic Caché consistent copies, two files are expected to be present on the production or source host(s). Depending on whether the application is started on the mount host, a third file is expected on the mount host which can be used to shut down the application. Callout scripts have **no extension** (on Unix platforms), require **executable** permissions, should start with the type of script being used, i.e. “**#!/bin/sh**,” or perhaps “**#!/usr/bin/ksh**” and are named as follows:

- Freeze callout - **appsync_freeze_filesystem_<service plan or first generation copy label name in lower case>**
- Thaw callout - **appsync_thaw_filesystem_<service plan or first generation copy label name in lower case>**
- Unmount callout - **appsync_unmount_filesystem_<service plan or first generation copy label name in lower case>**

Since AppSync has two types of workflows, the files must be named appropriately. The only difference is in the name of the file, as noted previously – service plan versus repurposing workflow. When using service plans, the file name contains the service plan name, and when using the repurposing workflows, contains the 1st generation label name which is user created as well. Callouts are not utilized during the 2nd generation copy processing, just as a 2nd generation copy source is not quiesced by utilizing any of the built-in application APIs within AppSync – as AppSync does not freeze/thaw the 1st generation copy when creating the 2nd generation copy.

Each file should contain the commands to freeze, thaw, or shutdown the application, as appropriate. These scripts are generally provided by the application vendor. The following are rules for the scripts which are to be executed:

- The freeze/thaw files should reside on the production/source host, while the single unmount file to shut down the database before a refresh operation, should reside on the mount host, all residing in **/var/opt/emc/appsync/scripts** on the respective hosts
- The default timeout value for callout scripts is **20 minutes** - the callouts executed must complete within 20 minutes or will fail
- AppSync requires a return code of zero (0) from the called script, in order to continue. Return codes other than zero (0) are considered a failure and will cause the AppSync job to fail (unlike pre/post copy scripts)
- The scripts run as the user profile used when installing the product, which may be root or a sudo user (please see the *AppSync User and Administration Guide* for more details)

CALLOUT EXAMPLE

1. Create a service plan called **Prod-Bronze-Plan**, with the appropriate settings
2. Create two files, which are based on the service plan name – callout file names must always be lowercase:

```
appsync_freeze_filesystem_prod-bronze-plan  
appsync_thaw_filesystem_prod-bronze-plan
```

} Notice the SP name is made lower-case.

3. Add the contents of the script to be executed, to either file, such as:

```
$ cat appsync_freeze_filesystem_prod-bronze-plan  
#!/bin/ksh  
su -c /epic/prd/bin/instfreeze epicadm
```

```
$ cat appsync_thaw_filesystem_prod-bronze-plan  
#!/bin/ksh  
su -c /epic/prd/bin/instthaw epicadm
```

4. Place both files on the production host, or all nodes when using PowerHA clustering, in the following location:

/var/opt/emc/appsync/scripts

UNMOUNT CALLOUT

Callout scripts support freezing and thawing the Epic Caché application. The **instfreeze** and **instthaw** scripts, which Epic supplies, are incorporated into AppSync's job logic by utilizing the two different **callout** scripts; for creating consistent copies. Additionally, after AppSync mounts a copy, a **post-mount script** can be utilized to start the application. The post-mount script is configured within the UI. When doing so, the **unmount callout script** must also be utilized to ensure the mounted application is properly shutdown before being unmounted and refreshed, or expired. Please see the [Post-Mount Script](#) section for more details.

- The predefined location is the same as the freeze and thaw callouts – detailed below
- The script is called just before the file system gets unmounted:
 - Included as part of both “**unmount previous copy**” and “**unmount copy**” phases as depicted in **Figure 1**
 - Runs when “**unmount**” is run on demand as well
 - Will **not** run during “**unmount**” as part of the cleanup, after a failed mount operation

CREATE CACHÉ COPIES

Configuring AppSync to support Epic Caché starts with adding the hosts, the service label IP/Name if applicable, and configuring the callout scripts. The following sections help outline the next steps in the copy creation process - using service plans and/or repurposing workflows to initiate those copies.

SERVICE PLANS

The primary advantage of a service plan is its usefulness in creating multiple copies for multiple environments, repetitively. Service Plans can be configured to be limited by the number of copies by setting the expiration setting (default is **7**), choose the storage technology preference setting, such as choosing between the local array and RecoverPoint copies, as well as, but not limited to, choosing the mounting options. Any environment **subscribed** to the configured service plan, receives all the same settings, however, schedules and some mounting options can be configured with **override** settings. Additional detail on the schedule and mount override options can be read in the *AppSync User and Administration Guide*.

Service plans are often configured to provide reoccurring copies, which are maintained, either on a set schedule or on-demand. The settings are configured by first creating a service plan based on an existing template, or just by modifying one of the existing templates, though the former is preferred. Service plans are discussed in more detail in the *AppSync User and Configuration Guide*. For the purposes of this white paper, however, some relevant notes are worth discussing.

FILESYSTEMS SERVICE PLAN FOR EPIC CACHÉ

In order to support Epic Caché applications, with service plans, the AppSync **FileSystems** service plans are utilized; incorporating the appropriate callout scripts for application consistency. Configuring this type of service plan is initiated by navigating to the **FileSystems Service Plans** configuration menu, as seen in **Figure 2**. Please note, as mentioned previously, callout scripts are not configured within the service plan menu settings – please refer to the [AppSync Callouts](#) section for more details.



Figure 2 - FileSystems Service Plan

CONFIGURING THE FILESYSTEM SERVICE PLAN

Details for each service plan setting can be found, in depth, in the *AppSync User and Administration Guide*, however, some notes are more applicable when focusing on Epic Caché support. These include what to subscribe, callout scripts, what mount options are available and preparing for restore procedures.

- Ensure all file systems are subscribed to the same service plan, in order to protect them together. Knowing volume group layout is also necessary, and the entire volume group must be managed together. Subscribe the environment by navigating to **Copy Management**, selecting the host, or PowerHA service label IP/name, selecting all the file systems, click **Protect**, then **Subscribe to Plan**, and select the plan name.
 - When subscribing Epic Caché environments clustered under AIX PowerHA, use the service label IP/Name
- Callout scripts, as mentioned in more detail in the section [AppSync Callouts](#), are required for application consistency
- Filesystem service plans offer three types of mounting options (see the *AppSync User and Administration Guide* for more details):
 - **Default Path** – the default option - mounts Unix based copies to **/appsync-mounts**. This is considered a mount point, and can be configured to be, for example, **/mymounts** by overwriting the words “Default Path.”
 - **Mapped Path** - may be a better suited option for Caché, as a user can change the directory tree entirely, such as taking **/epic/prd01** and mounting it as **/epic/sup01**, as seen in **Figure 3**

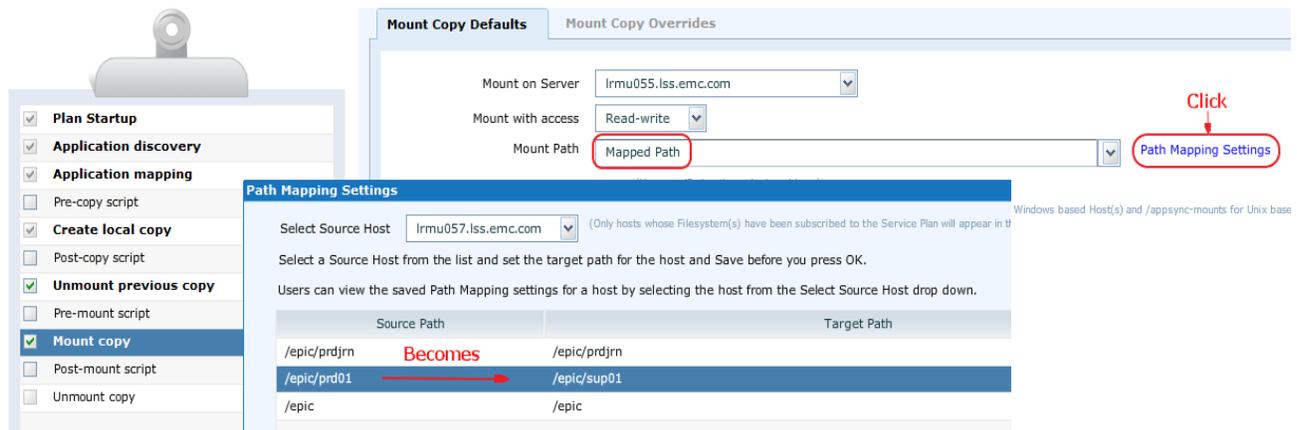


Figure 3 - Path Mapping Options

- **Same as original path** – only applicable if the mount host differs from the source, or any of the source cluster nodes
 - Both the **Mount Path** and **Mapped Path** options allow multiple copies to be mounted to the same mount host, since they will be presented in different ways.
 - Nested mount points, that is to say, mount points which point to a sub-directory of a parent file system, are supported. Please refer to the *AppSync User and Administration Guide* for more details.
 - Mounting to a PowerHA clustered environment is not supported, either back to the source or an alternative cluster
 - If mount operations are not specified in the service plan settings, and later the copies need to be mounted, please see the section titled [On-Demand Mounting](#).
- Restoring to a PowerHA clustered environment does not automatically make the volume group/file systems concurrent, they must be manually reconfigured to make the volume group concurrent again

RUNNING SERVICE PLANS

Running service plans can be accomplished in one of three ways, combined when subscribing through copy management, separately by running the service plan using the service plan menu, after subscribing, or creating a copy of a single file system, which also subscribes it to the service plan for later copy management. There are differences between the three, which are depicted below. The

one main advantage of running the process when subscribing, which is the first option, is if there are other environments subscribed to the same service plan, only the newly subscribed environment is run.

All three options are initiated by first navigating to **Copy Management > FileSystems**, and then selecting the host or the service label IP/name, when working in a PowerHA environment, where the file systems being subscribed are.

- Use the **Protect > Subscribe to Service Plan and Run** to subscribed to a service plan and run autonomously immediately
- Use the **Protect > Subscribe to Plan** to either allow the environments to run per the set schedule or run with the group already subscribed via on-demand, using the service plan menu
- A third option is to drill down to the individual file system, and click **Create copy using plan**. This runs the service plan autonomously against just that particular file system, which also subscribes it to the particular service plan

REPURPOSING WORKFLOWS

Similar to service plans, repurposing workflows offer the ability to provide different types of copy options, such as creating application consistent local or remote array copies, automate mounting, as well as scheduling options. Repurposing is different, however, in the sense that it runs against a single application, and does not utilize a copy count rotation methodology. This is to say, that AppSync provides a workflow to replicate an environment, consistently by utilizing callout scripts, for each environment - singularly. The repurposing workflow is not generally utilized for backup, rather for replicating to a test or development environment. More details on the repurposing workflow can be referenced in the *AppSync User and Administration Guide*.

Service plans offer a template which many different environments can utilize, where repurposing workflows are tailored per environment, with only a refresh option. A repurposing workflow is the only way an array based copy can be created from a RecoverPoint bookmark, such as when needing to create a remote array based copy (at the disaster recovery site for example) for a long term disaster recovery or development use case. This alleviates having testing or development use cases be performed against the RecoverPoint image access enabled bookmark, or simply provides a means to keep an extended copy of production at the remote site.

With the repurposing workflows, multi-generation copies can also be created. This is to say, a 1st generation copy, or one copy removed from the source, can be created, and then optionally an additional copy, termed as the 2nd generation copy, or a copy that is twice removed from the source, can be created. Please see **Figure 4** for a visual diagram overview and **Figure 5** for a visual representation of creating a remote array copy off of a RecoverPoint bookmark – when using SRDF, there is no need to use RecoverPoint as an intermediate step.

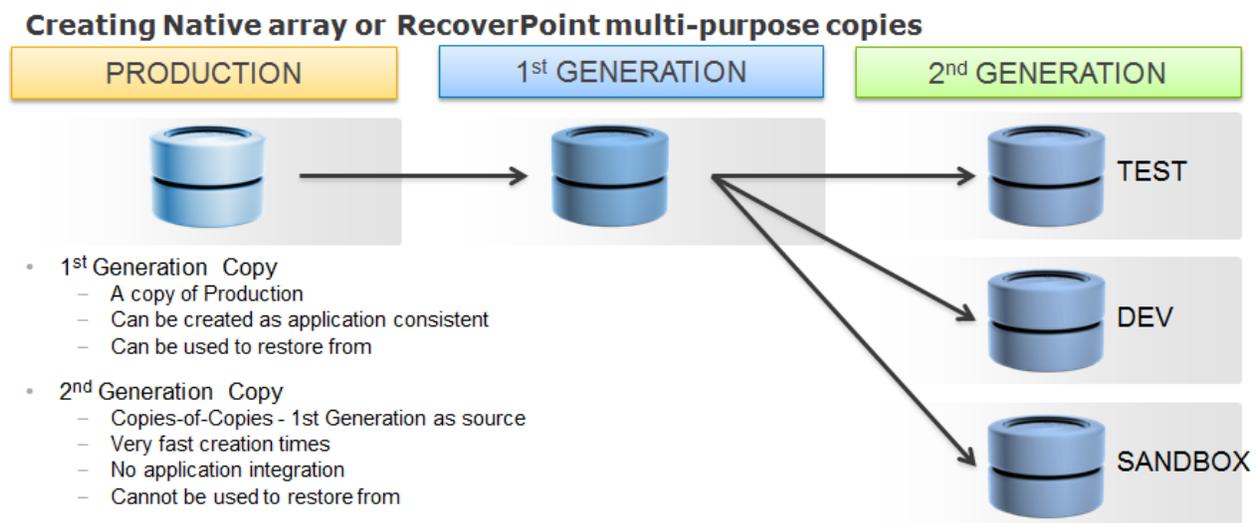
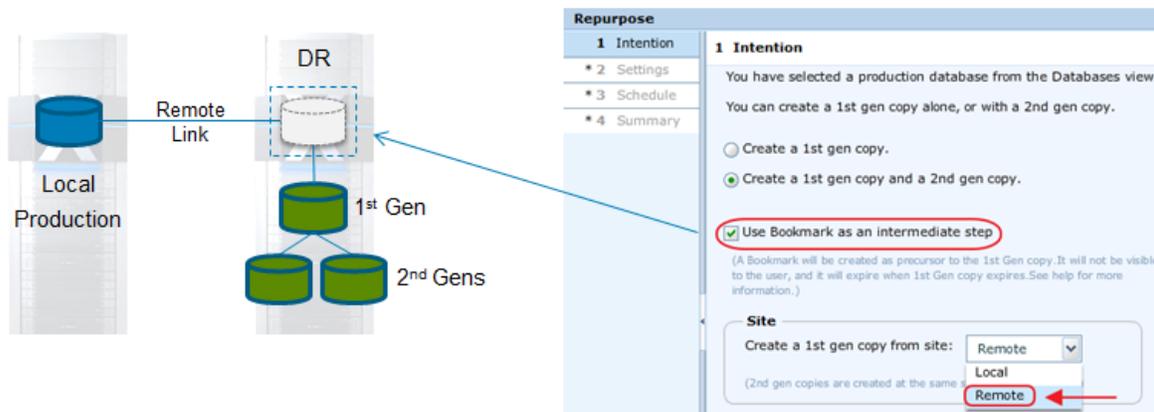


Figure 4 - Repurposing Overview



- Select **Use Bookmark as an intermediate step**, and then select **Remote**, in order to create remote array copies off RecoverPoint
- Both the local and remote native arrays must be added to AppSync
- **Remote** is also supported for SRDF

Figure 5 - Remote Repurposing

There are many use cases for creating multi-generation copies, two common scenarios are depicted below.

- Create a 1st gen copy (a “gold” copy left unmounted – can be used for recovery purposes), and create multiple 2nd gen copies (same time stamp) to development environments - alleviates having to interact with production multiple times
- Create a 1st gen copy, mount, alter data (data masking), unmount then create multiple 2nd gen copies for developers (without the sensitive information)

REPURPOSING MOUNT

Just like the service plan mount options, repurposing workflows offers the ability to either mount to the **Default Path**, **Mapped Path**, or **Same as original path**. Both the 1st generation and 2nd generation copies can be mounted, though when creating 2nd generation copies, the 1st should not be mounted at the time of the 2nd generation copy, otherwise, the 2nd generation copy will not be consistent. Application consistency is only achieved during the 1st generation copy. In some cases, however, the 1st generation copy is first mounted to mask sensitive data, which should then be unmounted before creating the 2nd generation copies.

Often, however, the repurposing workflow is created and no mount options are specified at the time the job is run. In this scenario, the mount operation is considered an on-demand mount, and is discussed in more detail, in the [On-Demand Mounting](#) section.

REPURPOSING REFRESH

Refreshing a repurposed copy, with file systems, differs from how repurposed copies are refreshed for SQL or Oracle. Since multiple file systems are repurposed together, it is through the **Copy Management > FileSystems**, and clicking on the **Repurposed Copies** button at the bottom of the screen. This menu, as seen in **Figure 8**, offers the ability to refresh the file system group. Properly labeling the repurposing copy is therefore critical to distinguish between environments.

ON-DEMAND MOUNTING

Not every workflow will be configured ahead of time to mount, or perhaps a different mounting operation must take place that is in response to, for example, a mount host not being available. On-demand mounts are activities where a copy is mounted through a wizard, or through a set of REST APIs or CLI commands. This white paper will only address the UI on-demand mount operations.

Service plan on-demand mounts involve a process where the copy, seen through the service plan’s **Copies** menu, as seen in **Figure 6**, is mounted by selecting one of the file systems, and then clicking **Mount**. Only one file system is selected, at this point, but in the second step of the wizard, as seen in **Figure 7**, any other file system that was created at the same time, can optionally be selected. In most all cases, the additional file systems should be selected and mounted together, thus providing a complete replica.

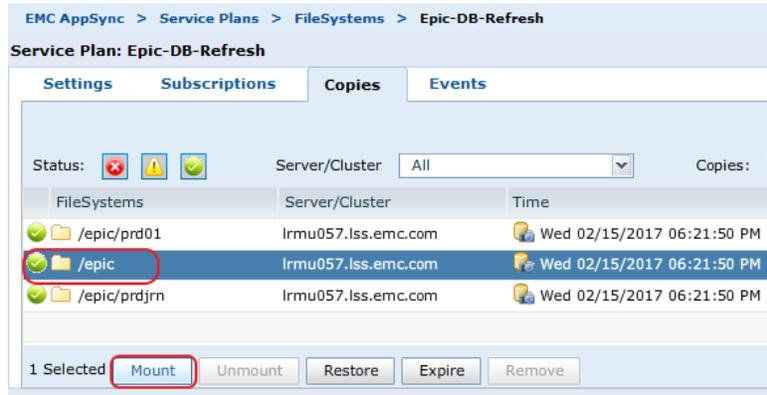


Figure 6 - Service Plan On-Demand Mount

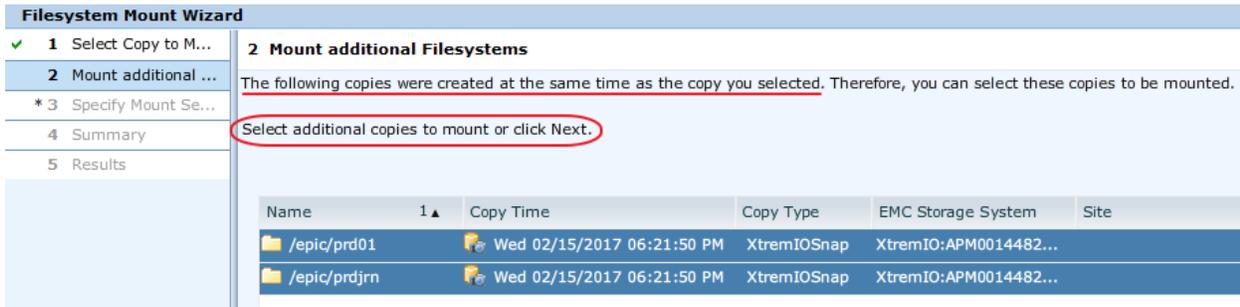


Figure 7 - Selecting Additional Filesystems

Repurposing workflows offer two methods for on-demand mounting. The easiest method is similar to when needing to refresh the repurposed copy manually, and that is by navigating to **Copy Management > FileSystems**, and clicking on the **Repurposed Copies** button at the bottom of the screen. This menu, as seen in **Figure 8**, offers the ability to not only choose the group to mount, but also offers the ability to refresh, or modify the repurposing settings, by clicking on the **Repurpose** link, for each group.

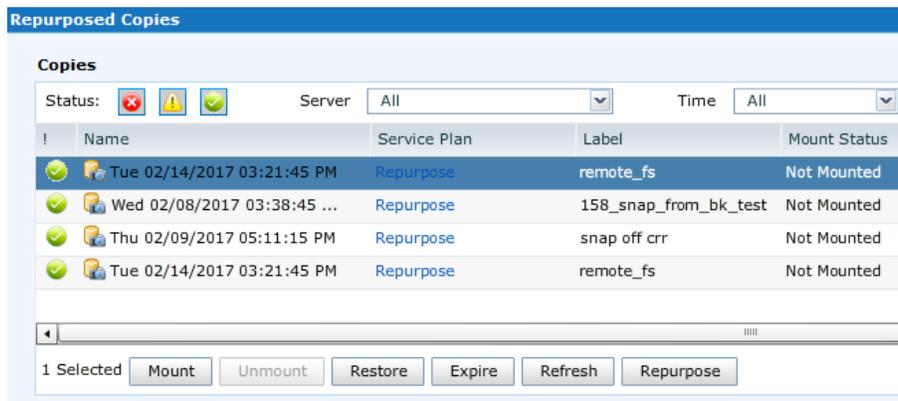


Figure 8 - Repurposed Copies Menu

The second way to initiate an on-demand mount of a repurposed copy is similar to the service plan method, by simply clicking the copy created previously, and then selecting mount, however, repurposing on-demand mounts start by navigating to **Copy Management > FileSystems**, selecting the node or service label IP/Name, selecting one of the file systems, clicking on the copy and selecting **Mount**, as seen in **Figure 9**.

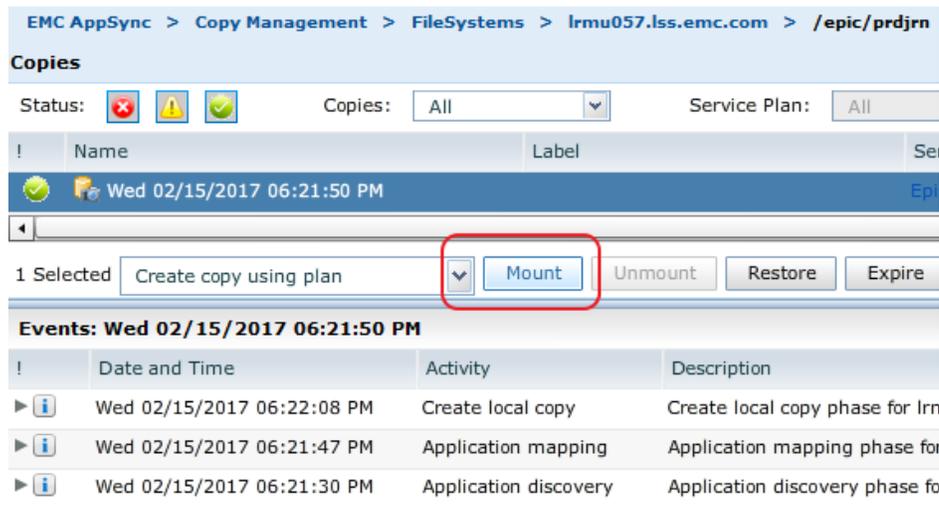


Figure 9 - Repurposing On-Demand Mount Option Two

POST-MOUNT SCRIPT

As mentioned previously in this white paper, callout scripts are utilized to create consistent copies of the Epic Caché environment. Service plans do, however, have other scripting options. While callouts support the copy process, other scripting options may also be desired, such as using a post-mount script. In some cases, after mounting the copy to a host, a post-mount script may be utilized to start the application, providing a completely robust and automated solution. The workflow can be seen in **Figure 10** depicted below.

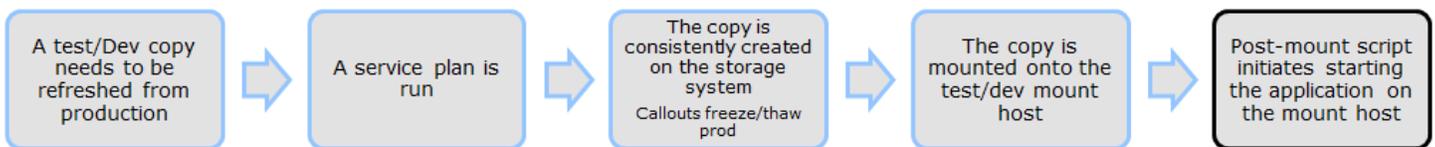


Figure 10 - Service Plan with Post-Mount Script

When considering the use of a **post-mount script** to start the application on the mount host, ensure an **unmount callout script** is also utilized; in order to shut down that application before the service plan is run again, providing the workflow as seen in **Figure 11** below.

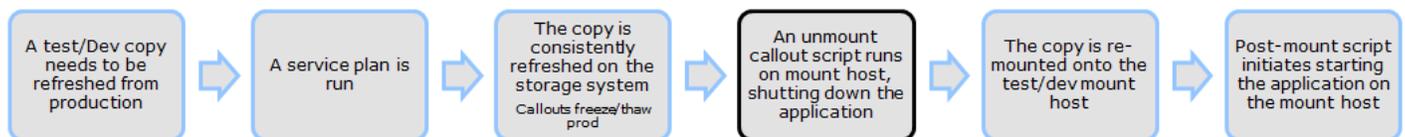


Figure 11 - Service Plan with unmount callout script

As seen in the second workflow, an **unmount callout script** is utilized to properly shutdown an environment that is started with a **post-mount script**. This ensures the application is shutdown properly before being refreshed. Unmount callout scripts, as all callout scripts, are discussed in more detail in the *AppSync User and Administration Guide*.

VPLEX INTEGRATION

Integrating with VPLEX is a matter of qualifying the backend storage arrays, and knowing which options to configure. Managing an Epic Caché environment with AppSync, adds no additional restrictions. This is to state that following both the *AppSync Installation and Configuration Guide* and the *AppSync User and Administration Guide*, to configure VPLEX, will support Epic Caché within the existing VPLEX support scope. All workflows that are supported for file system service plans and/or repurposing workflows are supported for Epic Caché, such as choosing which cluster and/or leg to create the copy on, as well as the three mounting options: Native array, VPLEX virtual volume mount (default), and Enable VMware cluster mount.