



**HIGHLIGHTS**

- Choice of object and file-based replication for a comprehensive solution
- Object-based is 26 times faster with Incremental Replication
- File-based replication allows granular file-level recovery
- Protects data against disasters with highly efficient replication to DR site
- Quick failover capability minimizes recovery times
- Choice of object and file-based replication for a comprehensive solution.
- Replicating source system to multiple targets maximizes coverage
- Archiving is easier by keeping snapshots longer than on source system maintaining CIFS/NFS access points and storage quotas across replications improves DR capabilities
- Choice of GUI or CLI interfaces offers flexibility to administrators

# BlueArc® JetMirror™

## High Speed Replication for Disaster Recovery at Scale

### Intelligent Architecture

BlueArc JetMirror uses an intelligent object-based architecture to achieve high speed replication. An object-based file system creates a collection of sub-file data containers which includes data objects, the master index objects and metadata objects like file access control and directory structure. The architecture organizes data as a collection of unique objects, instead of a complex hierarchy of files, folders, directories. This provides considerably flexibility to perform data and metadata operations independently and more efficiently, resulting in higher performance with less overhead on the CPU and storage. For example, metadata updates are done without opening the data object for a large file. On the other hand, with traditional file systems the data and metadata associated with a particular file are kept together in the same file.

### High Speed Replication

Asynchronous data replication is the standard method for moving large amounts of data, especially in data protection and disaster recovery applications. This process allows lower bandwidth connections to reduce costs and enables scheduling of data transfers to fit business needs. When the replication job begins, JetMirror doesn't spend cycles accessing file directory structures in preparation for a data transfer; it simply sends the objects to be replicated and the master index. Then, the target system can reassemble the files, the file system structure, and security from the objects sent. While file-based replication requires walking through directory trees and opening files to read the metadata to select files for replication the object-based replication simply scans through the indexed list of objects looking for objects that have not been replicated based on a date code on each object, and replicates the object to the target system resulting in substantial reduction in overhead. If only one object associated with a single file has changed, only that object is replicated, not the body of the file which will already be present on the target from a prior replication. Tests show that object-based replication of incremental files is 26 times faster than traditional file based replication.

**Highly Efficient Data Replication**

JetMirror with its object based replication is considerably more intelligent than “block” replication solutions, and in most cases even faster than block replication. Block- or volume-based replication essentially parses a data set into sub-file blocks and has no awareness of the content of each block, which means that ‘empty’ blocks, like white space in a photo, will consume processing and storage capacity the same as those which actually contain data. JetMirror’s object-based replication also shares the advantage of sub-file granularity, but has the content awareness to skip empty blocks.

JetMirror also offers the ability to maintain multiple snapshots on the remote site, more than on the primary site, and provide deeper backup of data. Other block replication solutions prevent this type of value add by requiring the file system to be exactly the same at both ends since they are essentially doing an image replication.

**Disaster Recovery: Easy-to- Manage, Quick Failovers**

JetMirror’s disaster recovery capabilities based on object replication quickly brings data back online quickly and efficiently through enhanced tools that automate system failover and failback. Administrators minimize unplanned outages and save thousands of dollars. Routine system maintenance tasks can be performed while the data is online, further minimizing downtime.

If disaster strikes, the target filesystem is promoted to source filesystem. Administrators can rollback to a specific snapshot, and/or simultaneously initiate failover for multiple filesystems. The source system which handed over its lead role to the target filesystem will be automatically updated by reverse replicating any changes that occur while the target system is in the lead.

**Fast DR Startup**

Object replication maintains the replication status on both the source and target filesystem. If the replication relationship is broken, such as during a system shut-down or move, when the relationship is re-established, incremental replication can continue, rather than requiring a full re-sync of the filesystem.

**DR System as Archive**

With object replication, the source and target filesystem can be different sizes, which allows the target filesystem to retain multiple snapshots on the target filesystem and for longer than on the primary filesystem. This greatly increases the ability to use the DR system as a longer term archive.

**Multi-Target Replication**

A source filesystem can be included in multiple replication schedules, allowing for multi-target replication. The snapshots and replication states for each replication job are separately maintained.

**Quotas Maintained Across Replication**

As with filesystem and directory structures, quotas are stored as objects which are replicated with the filesystem. Hence quotas are maintained across replication.

**CIFS and NFS Access Points Maintained**

Access points (whether CIFS share or NFS exports) are contained in the registry object which is stored in a special location on filesystem. The registry object is included in the replication job, allowing for enhanced DR for failover and failback. In case of disaster when the target filesystem is promoted to source level, CIFS and NFS access point information is available automatically.

**File Replication**

JetMirror also supports file replication which allows it to recover a more granular subset of files compared to object replication which works only at the filesystem level. The additional file replication capability comes in handy when mirroring is used as an additional data protection layer in conjunction with backup rather than just as a standalone Disaster Recovery System.

FEATURES	
Object Replication	Stores core structures and user data as objects, rather than files.
High Performance	Object-based replication improves full replication performance 2-3 x and incremental replication 26x compared to file-based replication. Benchmarked at speeds of 600-900MB/s on high speed networks.
Quick Failover	Enhanced tools automate system failover and failback
Archival	Use the DR system as archive by storing snapshots for longer periods on the target system than source
Multi-Target Replication	Source can be replicated on multiple targets for maximum safety
Ease of Management	CIFS/NFS Access Points and Quotas are maintained; both CLI and GUI options are available for management
SYSTEM REQUIREMENTS	
BlueArc Software	BlueArc System Software 8.0 or later
Network Protocol	NFSv3, NFSv4
VMware Applications	VMware vCenter Server and Client
File System	WFS-2 for Object and File Replication WFS-1 for File replication
OPTIONAL LICENSED FEATURES	
BlueArc JetCenter	To simplify data management and protection with VMware vCenter
BlueArc JetClone	To create BlueArc managed writeable clones



**BlueArc Corporation**  
 Corporate Headquarters  
 50 Rio Robles  
 San Jose, CA 95134  
 t 408 576 6600  
 f 408 576 6601  
 www.bluearc.com

**BlueArc UK Ltd.**  
 European Headquarters  
 Queensgate House  
 Cookham Road  
 Bracknell RG12 1RB, United Kingdom  
 t +44 (0) 1344 408 200  
 f +44 (0) 1344 408 202