

# Tiger BRIDGE

## Tiger Bridge 4.1.10 Release Notes

<i>What's New in Version 4.1.10</i> . . . . .	2
<i>Fixed Known Issues in Version 4.1.10</i> . . . . .	2
<i>Upgrading to Tiger Bridge 4.1.10</i> . . . . .	3
<i>Unresolved Known Issues</i> . . . . .	4

This document provides release information for Tiger Bridge version 4.1.10. It discusses new features and unresolved known issues in this release.

## What's New in Version 4.1.10

### Support for Hitachi HCP

You can use Hitachi HCP as a replication target. For prerequisites and configuration steps, refer to the Tiger Bridge 4.1.10 Administration Guide.

### Support for Seagate CORTX

You can use Seagate CORTX as a replication target. For prerequisites and configuration steps, refer to the Tiger Bridge 4.1.10 Administration Guide.

### Data Replication Policy and Synchronization of Other File Operations

With Tiger Bridge 4.1.10 the synchronization of the contents of source and target is no longer governed by the data replication policy. Thus, if you delete or rename a file on the source, the respective operation is synchronized on the target immediately and not after the timeout specified in the data replication policy elapses.

## Fixed Known Issues in Version 4.1.10

### NAS Source Fixes

Tiger Bridge 4.1.10 includes the following improvements and fixes on a NAS source:

- Tiger Bridge automatically re-connects to a NAS source, to which it has temporarily lost connection
- fixed slow replication, when the control folder of a NAS source contains read-only orphan files
- fixed reclaiming read-only files on an NFS share source.
- Tiger Bridge automatically detects, when a folder on a network share source is renamed.
- after restarting the Tiger Bridge service (on applying configuration changes, for example), it automatically resumes receiving notifications from NAS sources.

## Replicating Bigger Files to S3 or S3-compatible Target

Version 4.1.10 resolves a problem, which could result in failed replication of a file with size of 500 GB or above over a slower connection to Amazon S3 target or any other target using the S3-protocol.

## Removing Replicated Folders from the Target

With version 4.1.10, whenever data is deleted from the source, Tiger Bridge automatically deletes from the target not only all deleted files, but also their containing folders.

**Important:** *This behaviour is valid only when you have configured Tiger Bridge's delete mode to remove the replica from the target, upon deleting the file from the source.*

## Replacing Replicated Files with Stubs When the Source is Full

Version 4.1.10 resolves a problem, which could prevent Tiger Bridge from replacing files replicated on the target, if there was not enough free disk space on the source. Tiger Bridge 4.1.10 is capable of freeing space on your source, even if there is no free space on the disk.

## Replicating Files with Large Security Descriptor

With Tiger Bridge 4.1.10 files can be normally replicated regardless of the size of their security descriptor.

## Improved Support for Manually Retrieving Files from Cloud Targets

Version 4.1.10 prevents possible corruption of partially retrieved files with size above 20 MB, when a user attempts to manually retrieve them from a cloud target.

## Improved Support for Files with Unicode Replacement Characters in Their Name

Tiger Bridge 4.1.10 provides improved support for Unicode file names, preventing cases when replication fails if the name contains a replacement character.

## Upgrading to Tiger Bridge 4.1.10

To upgrade Tiger Bridge to this new version, you should simply run the installation of version 4.1.10 on the computer running Tiger Bridge. All configuration settings will be preserved after the upgrade.

## Unresolved Known Issues

### Retrieving Offline Files from Google Cloud

Unlike other cloud targets, offline files stored on a Google Cloud target are directly retrieved on the source when you attempt to open them or to manually rehydrate them.

### Using Versioning Software on Azure Append/Page Blob

When using versioning on Microsoft Azure append or page blob as a target, you should keep in mind that the first version of each file is not kept and the second version overwrites it. From the second version onwards versioning works as expected on Azure append and page blobs.

A workaround to the problem is to introduce an insignificant change to the file after it has been initially replicated on the Azure append/page blob (such as an added interval at the end of a text document, for example) in order to trigger versioning for that file from that change onwards.