TIBCO ActiveSpaces®

Release Notes

Version 4.7.0
September 2021

Contents

About This Product

The TIBCO ActiveSpaces® software is a distributed in-memory data grid product. Some features of ActiveSpaces® include use of familiar database concepts, high I/O capacity, and network scalability.

ActiveSpaces features a complete redesign and reimplementation of the product and is straightforward to understand, use, and administer.

Product Editions

ActiveSpaces is now available in two editions: Community Edition and Enterprise Edition.

	Community Edition	Enterprise Edition
Ideal for	Getting started with ActiveSpaces for implementing application projects, including proof of concept projects, for testing, and for deploying applications in a production environment. Production deployments running up to 5 nodes (a total of the copyset nodes or proxies in your data grid) For more information, see Terms used in Community and Enterprise Editions.	All application development projects, and for deploying and managing applications in the production environment of an enterprise. Production deployments with more than 5 nodes (a total of the copyset nodes or proxies in your data grid) For more information, see Terms used in Community and Enterprise Editions.
Features	All features of the Enterprise Edition except enterprise monitoring using dashboards.	Includes all features presented in this documentation set.
Limitations	Run up to 5 nodes (a total of the copyset nodes or proxies in your data grid). Although the community license limits the number of production instances, you can	No limitations on a total of the copyset nodes or proxies in your data grid.

	easily upgrade to the enterprise edition as your use of ActiveSpaces expands.	
Cost	Free	Paid
Compatibility	Compatible with both the enterprise and community editions of TIBCO FTL®	Depends on the enterprise edition of TIBCO FTL for monitoring and management of data grid components and secure communication.
TIBCO Support	No access to TIBCO Support	Access to TIBCO Support

Terms used in Community and Enterprise Editions

- Node - a copyset node or proxy where each copyset node or proxy is an operating system process with a unique process ID.
- Process ID - For the purposes of the definition of Node, Process ID means a standard computer industry term that uniquely identifies each operating system process.
- Copyset - For the purposes of the definition of Node, “Copyset” means a logical grouping of nodes such that a portion of the data is shared uniformly by all the nodes that form a copyset.

Terminology Used to Address the TIBCO FTL Realm

With TIBCO FTL 6.1 or later, ActiveSpaces uses the realm service capabilities or processes of the TIBCO FTL server. The following changes are made to the terminology to generically address the components of TIBCO FTL 5.x and TIBCO FTL 6.x:

The Term Used in the Document	The Equivalent Component in TIBCO FTL 5.4.1	The Equivalent Component in TIBCO FTL 6.1 or Later
Realm service	Realm server	Realm service running on the TIBCO FTL server
Realm service URL	Realm server URL	TIBCO FTL server URL
Backup realm service	Backup realm server	TIBCO FTL server that is a member of a cluster of three or more TIBCO FTL servers
Primary Realm	Primary Realm Server and its Backup Realm Server	A cluster of primary TIBCO FTL servers that provide realm services for the data grid.
Satellite Realm	Satellite Realm Server and its Backup Realm Server	A cluster of satellite TIBCO FTL servers that are connected to a cluster of primary TIBCO FTL servers.

New Features

The following features have been added in this release of TIBCO ActiveSpaces®.

A script and Dockerfile are now included in the installer

A script and Dockerfile are now included in the installer under `build-images` to enable building more flexible and customizable ActiveSpaces® docker images.

These replace the docker-images from previous releases.

For more information, see "Building a Docker Image" in *TIBCO ActiveSpaces® Administration*.

The new verbose log level for tibdg and tibdgadmind

The new `verbose` log level is added to `tibdg` and `tibdgadmind` commands. Enable the log level by setting `--trace verbose` on the command line.

Added go.mod files

The `go.mod` files are added to the Go client library and sample.

Ability to run the ActiveSpaces® proxy as a TIBCO FTL service

Added ability to run the ActiveSpaces proxy as a TIBCO FTL service so that there is no need to configure an external host and port.

Improved grid selection in grid dashboard

The grid dashboard is updated to have a selector to choose data from a specific grid.

A new chart on Grafana dashboards

Added the following details to Grid and Node dashboards:

- Number of scans
- Number of rows scanned per second
- Number of rows deleted per second

The node activity dashboard is improved

The node activity dashboard is updated to include more panels that display information about disk and memory usage.

Implementation of automatic retries of primitive operations

If the client detects that its proxy or a copyset leader has failed during a primitive PUT, GET or DELETE operation, then the operation is retried automatically provided that the client's timeout has not reached.

No limit on length of a copyset name

The limit of eight characters imposed on the copyset name when printing the output from the tibdg status command is now removed.

Internal retry of operations when an out-of-sync secondary becomes an in-sync secondary node

Operations that previously would have timed out are now retried automatically in the following case so that the client does not experience the timeout:

When an out-of-sync secondary node becomes an in-sync secondary node with a primary node.

The --user-password-file can store obfuscated passwords

You can use the masking of a password feature from TIBCO® FTL 6.6.0 and higher.

For more information about how to generate a masked password, see the `--mask` option available in the "FTL Administration Utility" section in *TIBCO® FTL Administration*.

A proxy responds back to the client once the DDL statement is completed

Improved coordination between proxies and nodes so that a proxy only reports to the client once the DDL statement is completed by all running copysets.

Changes in Functionality

The following functionality and features have been changed in this release of TIBCO ActiveSpaces®.

Added the STATUS field when listing checkpoints

The `tibdg checkpoint list` command output has the following columns:

ID, NAME, DIRECTORY, TIMESTAMP, and STATUS

The STATUS field has one of the following values:

Value	Description
in progress	When you are currently in the process of taking a checkpoint
success	When the checkpoint was completed successfully.
failure	When the errors occurred while taking the checkpoint.
mirroring	when a checkpoint is being received by the mirror grid. The primary grid displays the status as `success`.

lz4 library update

The lz4 library is updated from version 1.8.3 to 1.9.3.

Improved performance for queries with an asterisk

Improved performance for certain queries which only contain an asterisk (*) character in their result list.

Improved memory usage for concurrent queries.

Memory usage in the node is more efficient when running concurrent queries or iterators for a sustained period.

Support for fixing secondary index issues at startup

Secondary nodes that performed bulk catchup at some point in their history could have written orphaned secondary index rows and written incorrect exact row counts.

These issues have been resolved, and in addition, tibdgnode processes do the following:

• Perform a one-time scan of all secondary indexes

• Repair incorrect secondary indexes and update exact row counts for those indexes

• Update primary indexes that have exact row counts enabled to ensure that all data is consistent and counts are accurate

Metadata no longer leaks from a node

After recovering from a grid rollback or setting a primary server event, the metadata does not leak memory in the node.

Optimized number of disk writes performed by tibdgkeeper process

Optimized number of disk writes performed by tibdgkeeper process during leadership changes.

Support for compressing checkpoint lists

The following grid configuration option is added to help compress checkpoint lists:

```
checkpoint_list_compression=(enabled|disabled)
```

Enabling this option causes a reduction in the size of the list of checkpoint-related metadata written to disk by the state keepers and nodes. By default, this option is enabled for new grids and requires no change in checkpoint recovery procedures.

For existing grids that are upgrading to this version, this feature is not enabled by default. To use this feature on existing grids, an administrator must first make sure that all the grid processes have been upgraded.

After all the grid processes are upgraded, run the following command:

```
tibdg grid modify checkpoint_list_compression=enabled
```

You do not need to restart the grid processes to see this change. Only checkpoints taken after this property is modified apply compression to the metadata.

Deprecated and Removed Features

No features have been deprecated or removed as of this version of TIBCO ActiveSpaces®.

Migration and Compatibility

This version of ActiveSpaces is backward compatible with ActiveSpaces 3.x. However, ActiveSpaces 3.0 and later is not backward compatible with ActiveSpaces 2.x.

For information about upgrading from an earlier version of ActiveSpaces, see "Upgrading from an Earlier Version" in *TIBCO ActiveSpaces Installation*.

This release requires TIBCO FTL® 6.4 or later. For information about migrating from TIBCO FTL 5.x to TIBCO FTL 6.4, see the "Migration and Compatibility" section in *TIBCO FTL® Release Notes* and "Upgrade Migration to a New Release" section in *TIBCO FTL® Administration*.

Note: For the upgrade to be successful, the latest version of ActiveSpaces `tibdg grid rebuild` must run on the existing data grid before upgrading the ActiveSpaces components.

After installing this version of ActiveSpaces, you can take advantage of the new Changes in Functionality.

Closed Issues

The following issues have been fixed in this release of TIBCO ActiveSpaces®.

Key	Summary
FFY-4169	A race condition can cause a node process to crash while handling many concurrent iterators and queries.
FFY-4098	The command `DROP INDEX IF EXISTS <table>.<index>` throws an exception even if the table does not exist.
FFY-4093	A proxy and node retry the state keeper at a slower interval than expected at startup.
FFY-4090	The Mac OS installer files are not signed.
FFY-4072	The node, state keeper, and proxy processes do not shut down gracefully upon receiving the `SIGTERM` signal.
FFY-4027	After adding a new copyset, the redistribution could time out while rows are being migrated to the new copyset and TTL/expiration also happens for a row at the same time it is being sent. The redistribution protocol incorrectly assumes there is no activity, resulting in a retry of the redistribution.
FFY-4023	When you take checkpoints after performing a Live Backup and Restore, records in an internal system table accumulate in nodes that are not in a gridset.
FFY-4022	Boolean configuration options with a default value `true` are not parsed correctly and can not be changed from their default values.
FFY-4005	The proxy crashes and generates a segmentation fault for a query when an alias has a different case than the case of a column name in a query.
FFY-4003	The `tidbdg grid delete <grid name>` command does not delete the grid.

Key	Summary
FFY-4002	Queries that are either very long or have ASCII control characters can corrupt the log output.
FFY-3999	A secondary node that is not synchronized can send a protocol message to a primary node. The protocol message then incorrectly resets the primary node.
FFY-3995	The `tibdgSession_ExecuteUpdate` function incorrectly allows SQL DDL statements to be executed on sessions that are created on a checkpoint. Now, the call to the `tibdgSession_ExecuteUpdate` function returns an error if the session is created on a checkpoint.
FFY-3992	A timing issue related to a secondary node that is synchronizing with the primary node while live operations are performed on the same key could lead to incorrect secondary index information for that key. That secondary index shows unexpected issues such as warnings during expiration, a deleted key still remaining in a locked state, and future operations on the key timing out.
FFY-3987	In certain cases, a node that is overloaded with operations can exit unexpectedly or goes into a state where it uses CPU excessively.
FFY-3980	A secondary node goes to an infinite loop while performing the catch up task on a small number of rows with large size. The secondary node then has high CPU usage and the secondary node fails to complete the catch up task.
FFY-3970	Operations performed on a table during a period of compaction fail if a table is first dropped and then recreated.
FFY-3964	Certain write patterns, particularly involving tables with TTL enabled, can cause some tibdgnode processes to use excessive amounts of disk space over time.
FFY-3963	The client becomes unresponsive forever if a node or a proxy fails during the execution of a query, and the query has fetch timeout set to 0.
FFY-3955	In some cases, slow disk writes in a primary node could affect the node's ability to handle read requests.

Key	Summary
FFY-3948	The proxy adds a warning message to the log when the client's DDL command is not valid.
FFY-3947	The count of currently active queries does not decrement properly in certain disconnect scenarios.
FFY-3937	A secondary node becomes a dead secondary node due to compaction.
FFY-3936	An ongoing background store operation prevents timely shutdown of the node, potentially even delaying transition from secondary to primary node in a copyset.
FFY-3935	A node's shutdown and reset time is longer than necessary. Now, a node discards pending work instead of processing it once a shutdown or reset is triggered.
FFY-3932	The node repeatedly attempts to perform mirroring when it has no method of contacting the mirror grid and generates the following warning: `Queue for thread group 'garbage' is at limit.`
FFY-3929	Some tibdg commands do not require a realm server connection but those commands still wait for up to 10 seconds for a realm server connection.
FFY-3927	The status of reindexing is not included in the output from the `tibdg status` and `tibdg node status` commands for tables that are created after the node is started.
FFY-3909	The component in a node responsible for coordinating checkpoints receives unwanted configuration update messages.
FFY-3907	A secondary node whose disk write operations are taking longer than 15 seconds could update its state to out of sync more than necessary.
FFY-3904	Issuing the `tibdg node status` command to a node after it is restarted during bin migration could cause the node to exit.
FFY-	In some situations, you cannot create an index on a table, even if the `CREATE TABLE`

Key	Summary
3902	command is successful.
FFY-3869	The mirroring process does not retry if it fails to record an internal record, which causes the mirroring process to fail. The mirroring component on secondary nodes responds to requests incorrectly and generates the following warning: `Sending MIRRORING COMPLETE with error: Can't open checkpoint.`
FFY-3866	The mirroring process scans through the entire journal metadata table to determine whether it can begin the process of sending rows to a mirror grid.
FFY-3865	After a redistribution, rows may not be cleaned up correctly in all cases.
FFY-3857	In some cases, the Go garbage collector destroys C objects that are still in use when freeing Go objects. Now, calls to `runtime.KeepAlive` are added to ensure that the Go objects are not finalized until after the C objects are used
FFY-3853	The state keeper does not report the bin assignments correctly after a bin redistribution.
FFY-3849	If multiple timeout errors happen during a redistribution, there is the possibility that the node in the new copyset does not retry and continue the redistribution after the timeout. The node needs to be restarted to continue the redistribution.
FFY-3841	A primary node with a slow disk does not correctly handle a case where the secondary node is repeatedly trying the same catchup and failing.
FFY-3840	A primary node could exit unexpectedly while catching up a secondary node.
FFY-3839	In a gridset, after the successful mirroring of a checkpoint to a remote mirror grid, the nodes of a primary grid could restart.

Key	Summary
FFY-3838	Occasionally, some delete operations fail to be mirrored to a mirroring grid.
FFY-3835	The tibdg proxy modify command does not modify the proxy's configuration.
FFY-3831	A secondary node could incorrectly handle a reply from primary during the catchup process, leading to unexpected warnings or behavior at the secondary node.
FFY-3830	The tibdg output does not list tables in an alphabetical order.
FFY-3822	When mirroring from a primary grid to a mirror grid, a large number of rows can cause the mirroring operation to fail at the mirror grid.
FFY-3807	When a node is restarted, there is a short period where full table scans might be used for queries instead of indexed scans.
FFY-3805	The documentation for the `TIBDG_STATEMENT_PROPERTY_DOUBLE_FETCH_TIMEOUT` property did not include a warning about using a zero value in a production system in the *TIBCO ActiveSpaces® API Reference* documentation.
FFY-3794	If a secondary node is not synchronized with the primary node and is performing the catchup operation, then it can severely impact live operations on the primary node.
FFY-3792	The `SQL DROP TABLE` command fails if the table's name includes a forward slash (/) character.
FFY-3789	The tibdg command takes too much time to execute a script used to create multiple tables.
FFY-3785	Secondary nodes send unnecessary data during mirroring.
FFY-3767	Compiling a Go application fails If the program includes both the Go API from TIBCO FTL and the Go API from ActiveSpaces due to a duplicate symbol being exported.

Key	Summary
	Now, the functions that are exported from ActiveSpaces Go libraries have tibdg in their names to prevent collisions.
FFY-3742	When a proxy receives multiple `CREATE TABLE` commands for the same table in rapid succession, every command after the first command throws the `table already exists` exception.
FFY-3739	When running simultaneous `CREATE TABLE IF NOT EXISTS` commands on the same table, some of the commands throw the `table already exists` exception even if the `IF NOT EXISTS` condition is specified.
FFY-3725	In some cases, nodes handle retries incorrectly when restarting and generates the following error: `txnleader got error in transaction abort whitelist op for unknown transaction.`
FFY-3723	The warning that prints in the node log when a memory limit is reached due to an `ORDER BY` query now displays the query string that caused it.
FFY-3711	The tibdg command `table stats <tablename>` returns the incorrect result when a primary node is unavailable.
FFY-3670	The node process can reset itself without proper log messages.
FFY-3633	A secondary node displays unexpected results in some circumstances for synchronized rows when the secondary node becomes a primary node and tries to synchronize those same rows to another node.
FFY-3607	When executing a sequence of commands by using a script, the tibdg command returns an error if none of the commands modify the grid configuration.
FFY-3377	The tibdg admin utility now prompts for a password when you start a data grid with authentication by using the following arguments: `-user <user_name> -password stdin`

Key	Summary
FFY-3235	Documentation is updated to specify how to create a node and a state keeper's data directory at custom location. For information, see "Defining a Data Grid" in *TIBCO ActiveSpaces® Administration*.
FFY-3205	In some scenarios, a primary node in a copyset unnecessarily triggers the synchronization process for the secondary node even if the secondary node is already synchronized.

Known Issues

The following issues exist in this release of of TIBCO ActiveSpaces®.

Key	Summary/Workaround
FFY-3493	**Summary:** Using `tibdgRow_ToString` does not display all columns when the SELECT list contains duplicate column names. **Workaround:** In the SELECT list, instead of column names, use the unique label names. For example, SELECT lastname, firstname, lastname AS ln2 FROM mytable.
FFY-2942	**Summary:** Queries involving GROUP BY do not work with expressions. The following limitations apply when using GROUP BY: • SQL CASE expressions cannot be used in select lists when aggregation is used. • SQL CASE expressions cannot contain aggregation functions. • Nested functions and nested expressions in select lists cannot be used when aggregation is used. **Workaround:** None
FFY-2711	**Summary:** The SQL `INSERT` parameter values must exactly match the type defined for the column in the table. No attempt is made to cast from the data type of the parameter to the data type of the column in the table. **Workaround:** Set the parameter value into the statement by using the type that exactly matches the column type of the table.
FFY-2140	**Summary:** Removing a mirror grid from a gridset, either by using the `tibdg gridset remove` command or by using the `tibdg gridset delete` command, results in an unexpected behavior if the `-p` or `--makePrimary` flags are not used. As a result, data grids cannot be added back after they are removed from the gridsets. **Workaround:** Use the `-p` or `--makePrimary` flag when performing an operation to remove a data grid from a gridset. Using these options transitions the mirror grid into a standalone grid. As a result, the data grid removed cannot be added to a gridset as a mirror grid.

Key	Summary/Workaround
FFY-1654	**Summary:** When creating a table using SQL, the SQL data types are mapped to their underlying FTL types. When the metadata for the table is returned, the original data types are not preserved and you can only see the following mapping: • long => BIGINT • string => VARCHAR • double => DOUBLE • datetime => TIMESTAMP • opaque => VARBINARY **Workaround:** For information about how SQL data types are mapped to ActiveSpaces data types, see the "SQL Data Type Mapping" section in *TIBCO ActiveSpaces® Administration* .
FFY-1569	**Summary:** In an iterator, calling `tibdgRow_ToString()` on a row or executing a query such as `SELECT * FROM <table>` does not always return columns in the order defined in the table. **Workaround:** Retrieve the columns individually by name and in the order you want.
FFY-1474	**Summary:** Occasionally, using special characters for identifiers is out-of-sync between the `tibdg` tool and the SQL DDL commands. In some scenarios, you can create a table by using the `TABLE CREATE` command, but not drop the table by using the `tibdg` tool or by using the `SQL DROP TABLE` command. **Workaround:** Create simple table names. The tables names must use lowercase characters, but no special characters or embedded spaces.

TIBCO Documentation and Support Services

How to Access TIBCO Documentation

Documentation for TIBCO products is available on the TIBCO Product Documentation website, mainly in HTML and PDF formats.

The TIBCO Product Documentation website is updated frequently and is more current than any other documentation included with the product. To access the latest documentation, visit https://docs.tibco.com.

Product-Specific Documentation

The following documentation for TIBCO ActiveSpaces® is available on the TIBCO ActiveSpaces® Product Documentation page:

- TIBCO ActiveSpaces® *Release Notes*
- TIBCO ActiveSpaces® *Installation*
- TIBCO ActiveSpaces® *Concepts*
- TIBCO ActiveSpaces® *Administration*
- TIBCO ActiveSpaces® *API Reference*
- TIBCO ActiveSpaces® *Security Guidelines*

How to Contact TIBCO Support

You can contact TIBCO Support in the following ways:

- For an overview of TIBCO Support, visit http://www.tibco.com/services/support.
- For accessing the Support Knowledge Base and getting personalized content about products you are interested in, visit the TIBCO Support portal at https://support.tibco.com.
- For creating a Support case, you must have a valid maintenance or support contract with TIBCO. You also need a user name and password to log in to https://support.tibco.com. If you do not have a user name, you can request one by clicking Register on the website.

How to Join TIBCO Community

TIBCO Community is the official channel for TIBCO customers, partners, and employee subject matter experts to share and access their collective experience. TIBCO Community offers access to Q&A forums, product wikis, and best practices. It also offers access to extensions, adapters, solution accelerators, and tools that extend and enable customers to gain full value from TIBCO products. In addition, users can submit and vote on feature requests from within the TIBCO Ideas Portal. For a free registration, go to https://community.tibco.com.

Legal and Third-Party Notices

THIS DOCUMENT COULD INCLUDE TECHNICAL INACCURACIES OR TYPOGRAPHICAL ERRORS. CHANGES ARE PERIODICALLY ADDED TO THE INFORMATION HEREIN; THESE CHANGES WILL BE INCORPORATED IN NEW EDITIONS OF THIS DOCUMENT. TIBCO SOFTWARE INC. MAY MAKE IMPROVEMENTS AND/OR CHANGES IN THE PRODUCT(S) AND/OR THE PROGRAM(S) DESCRIBED IN THIS DOCUMENT AT ANY TIME.

THE CONTENTS OF THIS DOCUMENT MAY BE MODIFIED AND/OR QUALIFIED, DIRECTLY OR INDIRECTLY, BY OTHER DOCUMENTATION WHICH ACCOMPANIES THIS SOFTWARE, INCLUDING BUT NOT LIMITED TO ANY RELEASE NOTES AND "READ ME" FILES.

This and other products of TIBCO Software Inc. may be covered by registered patents. Please refer to TIBCO's Virtual Patent Marking document (https://www.tibco.com/patents) for details.