

# **TRIPjxp v7.2-2**

## **Change History**

**2020-05-13**

Copyright (c) 2020 Smaser AG

Published By:  
Smaser AG, Bratustraße 9, 64293 Darmstadt, Germany  
E-mail [info@smaser.ag](mailto:info@smaser.ag), [www.smaser.ag](http://www.smaser.ag)



# End User License Agreement

All rights to this software, its documentation and logotypes of the TRIP product family and software (altogether "Software") supplied by Smaser AG (Smaser) are exclusively owned by Smaser.

The transfer of this Software, solutions or parts thereof requires the prior written agreement of Smaser. Furthermore, the customer has the right to use licensed Software and / or process solutions supplied by Smaser to the extent specified in his contract with Smaser.

The free-to-use non-commercial version doesn't require a prior written agreement with Smaser but such customers, organizations and/or third parties agree by using the software and / or solution of Smaser to be strongly obliged to keep all rights to this software, documentation and logotypes of the TRIP product family absolutely un infringed and protected.



# Table of Contents

About This Document.....	9
Version 7.2-2.....	9
New Features in v7.2-2.....	9
Check if a record has uncommitted changes [TRIP-5287].....	9
Corrected in v7.2-2.....	9
Record commit may not detect deleted field contents [TRIP-5280].....	9
Version 7.2-1.....	9
New Features in v7.2-1.....	9
Facet Baselines [TRIP-5129].....	9
Database name for TdbHistoryDetail and TdbSearchSet [TRIP-5128].....	9
Corrected in v7.2-1.....	10
Risk for connection failure with large facets.....	10
Version 7.2-0.....	10
New Features in v7.2-0.....	10
Save part of search history as procedure [TRIP-5111].....	10
DB designs from TdbDatabaseClusterDesign [TRIP-5110].....	10
New field fetch types.....	10
Enhanced support for threaded applications.....	11
Record fetch option can distinguish between databases for the same field	
name.....	11
HTML highlighting support for local use [JXP-107].....	11
Support for client-side use of TRIPviewX/TRIPcof.....	11
Corrected in v7.2-0.....	11
TRIPjxp could hang indefinitely on server-side crash [JXP-108].....	11
Client-side text extraction and HTML conversion under Linux/UNIX	
[JXP-110].....	11
Risk for array bounds violation in client-side text extraction and HTML conversion	
[JXP-111].....	11
Setting part id to retrieve a single part did not work [JXP-113].....	11
Array bounds exception thrown if a TdbSearchSet contained a deleted record	
[JXP-114].....	11
Access violation exception if not logged in to session [JXP-115].....	12
Version 3.1-0.....	12
New Features in v3.1-0.....	12
Graph search and navigation [JXP-70].....	12
New method TdbSearch.addSearchSet [JXP-97].....	12
TdbSession.isConnected [JXP-106].....	12
Corrected in v3.1-0.....	13
Version 3.0-8.....	13
New Features in v3.0-8.....	13
Copy method for TdbRecord [JXP-105].....	13



Corrected in v3.0-8.....	13
Committing records from a TdbLocalSession fails [JXP-103].....	13
Using XML special characters in names may cause exception [JXP-104].....	13
Version 3.0-7.....	13
New Features in v3.0-7.....	13
Corrected in v3.0-7.....	13
Upload of STRINGS larger than 2MB failed to DB w/o record names [JXP-102].....	13
Version 3.0-6.....	14
New Features in v3.0-6.....	14
Support for pessimistic record locking [JXP-100].....	14
Corrected in v3.0-6.....	14
INI-file argument to TdbTripNetSession constructor ineffective [JXP-99].....	14
Initialization of local TRIPcof use fails [JXP-101].....	14
Version 3.0-5.....	14
New Features in v3.0-5.....	14
Corrected in v3.0-5.....	14
Client-side use of TRIPcof/TRIPview-C not working under Unix [JXP-96].....	14
Use of TdbPoolInitializer could cause crash [JXP-98].....	14
Version 3.0-4.....	15
New Features in v3.0-4.....	15
Activity logging for connection pools [JXP-93].....	15
Overloaded acquire() method with timeout / blocking behavior [JXP-94].....	15
Session pool initialization [JXP-95].....	15
Corrected in v3.0-4.....	15
Risk for NullPointerException in TdbSession.licensedProducts() [JXP-92].....	15
Version 3.0-3.....	15
New Features in v3.0-3.....	15
Support the TRIPview-C product [JXP-85].....	15
NOBAF option for TdbDatabaseList [JXP-89].....	16
Corrected in v3.0-3.....	16
Documentation incorrect for TdbControlObject.getManagerRights [JXP-86].....	16
TdbControlObject.isFileManager returned incorrect value [JXP-87].....	16
TdbControlObject.isUserManager returned incorrect value [JXP-88].....	16
Data retrieval from databases with classification schemes failed [JXP-90].....	16
Version 3.0-2.....	16
New Features in v3.0-2.....	16
TdbTextExtractionInfo.ExtractFromStored property added [JXP-79].....	16
Use TdbStringField for text extraction via TextExtractionInfo [JXP-83].....	17
Corrected in v3.0-2.....	17
Invalid characters from text extraction not handled [JXP-77].....	17
Text extraction using TdbStringField may fail [JXP-78].....	17
Using TextExtractionInfo without setting Stream may fail [JXP-80].....	17
Text extraction update to named part records fails [JXP-81].....	18



TdbComponent.setName(String) fails [JXP-82].....	18
Version 3.0-1.....	18
New Features in v3.0-1.....	18
Keyword and metadata analysis [JXP-69].....	18
Use TRIPview for HTML conversion and text extraction [JXP-73].....	18
New method TdbSearchSet.delete() [JXP-74].....	18
Corrected in v3.0-1.....	19
Risk for NullPointerException in TdbStringField.convert() [JXP-75].....	19
Fetching large STRING values from part records [JXP-76].....	19
Version 3.0-0.....	19
New Features in v3.0-0.....	19
Client-side text extraction [JXP-46].....	19
Client-side HTML conversion [JXP-47].....	19
TdbControlObjectList.contains methods added [JXP-48].....	20
Session encryption [JXP-59].....	20
On-demand term lists [JXP-60].....	20
Facet classes [JXP-61].....	20
Access to database file location names [JXP-63].....	20
Retrieve control object for a single database [JXP-66].....	21
Synchronous indexing of databases [JXP-71].....	21
Synchronous indexing of single records [JXP-72].....	21
Corrected in v3.0-0.....	21
Extracting text from file with ampersand in the filename fails [JXP-55].....	21
Database language information missing from TdbDatabaseDesign [JXP-64].....	21
Chinese segmentation type "Word" not exposed [JXP-65].....	21
Version 2.1-6.....	22
New Features in v2.1-6.....	22
New method TdbSearchSet.delete() [JXP-74].....	22
Corrected in v2.1-6.....	22
Version 2.1-5.....	22
New Features in v2.1-5.....	22
Corrected in v2.1-5.....	22
TdbDatabaseDesign.getVifFile returns BAF file name [JXP-62].....	22
String field values may get deleted on update [JXP-67].....	22
TdbDateform throws exception on partial TRIP dates [JXP-68].....	22
Version 2.1-4.....	22
New Features in v2.1-4.....	22
Improve STRING field upload performance [JXP-58].....	22
Corrected in v2.1-4.....	23
Version 2.1-3.....	23
New Features in v2.1-3.....	23
Improve STRING field download performance [JXP-56].....	23
Corrected in v2.1-3.....	23
TdbException.getDebugMessage() not complete [JXP-57].....	23



Version 2.1-2.....	23
New Features in v2.1-2.....	23
TdbDatabaseDesign.getOwnerName method added [JXP-50].....	23
Filename field name for HTML conversion [JXP-53].....	24
Access rights via group memberships from TdbUserAccessList [JXP-54].....	24
Corrected in v2.1-2.....	24
Invalid exception from TdbSearchSet.clearCache() [JXP-51].....	24
No output to activity log if server exits prematurely [JXP-52].....	24
Version 2.1-1.....	24
New Features in v2.1-1.....	24
Support for search comforters [JXP-45].....	24
Corrected in v2.1-1.....	24
Version 2.1-0.....	25
New Features in v2.1-0.....	25
Open up for segmentation type "Word" [JXP-39].....	25
XPath as Query Language [JXP-40].....	25
XML Fragment Retrieval [JXP-41].....	25
Corrected in v2.1-0.....	25
TdbTuple.setValue(String,String) always throws exception [JXP-43].....	25
Exception when creating a record in DB with a record id field [JXP-44].....	25
Version 2.0-4.....	25
New Features in v2.0-4.....	25
Corrected in v2.0-4.....	26
Changing password for another user does not work [JXP-37].....	26
Changing password for current user does not work [JXP-38].....	26
Version 2.0-3.....	26
New Features in v2.0-3.....	26
Control object method for checking if a database is XML enabled [R1460].....	26
Enable custom ASE calls for insert and update of records [R1457].....	26
Corrected in v2.0-3.....	26
Version 2.0-2.....	26
New Features in v2.0-2.....	26
Automatically close idle sessions in connection pool [R1415].....	26
Corrected in v2.0-2.....	26
Version 2.0-1.....	27
New Features in v2.0-1.....	27
Finer control when specifying what to fetch from a TEXT field [R1411].....	27
Corrected in v2.0-1.....	27
Version 2.0-0.....	27
New Features in v2.0-0.....	27
Method TdbDatabaseClusterDesign.remove() added [R1404].....	27
Get field value size [R1372].....	27
Access field template collection via TdbRecord [R1346].....	27



FieldCount property for TdbDatabaseDesign [R1345].....	28
New class: TdbSearch [R1344].....	28
Support for thesaurus-based searching in TdbRecordSet [R1333].....	28
Access record timestamp through TdbRecord [R1316].....	28
Record ID available for new records [R1305].....	28
TdbDateform capable of formatting TRIP time stamps [R1300].....	28
Session isolation for connection pools [R1291].....	28
Support for TRIPxml 2.x added [R1124].....	28
Corrected in v2.0-0.....	28
Exception thrown when saving database cluster design changes [B1403].....	28
Base64 routines produce invalid encoding [B1294].....	29
Version 1.3-4.....	29
New Features in v1.3-4.....	29
Access to the did-you-mean facility [R1256].....	29
Corrected in v1.3-4.....	29
Less-than sign in command text caused corrupt XPI request [B1250].....	29
Version 1.3-3.....	29
New Features in v1.3-3.....	29
Corrected in v1.3-3.....	29
Months incorrectly parsed in TdbDateform.formatTimestamp [B1221].....	29
Risk for exception in TdbDatabaseAccess.get for empty scopes [B1244].....	30
Version 1.3-2.....	30
New Features in v1.3-2.....	30
TRIPgrid added to TdbProduct enum [R1199].....	30
Corrected in v1.3-2.....	30
Record name not parsed for databases not owned [B1214].....	30
Version 1.3-1.....	30
New Features in v1.3-1.....	30
Corrected in v1.3-1.....	30
Removal of all values from a field not working [B1182].....	30
Version 1.3-0.....	30
New Features in v1.3-0.....	30
Tuple lists [R1152].....	30
New method TdbField.getValueCount [R1158].....	31
Value insert method for TdbField [R1159].....	31
Value remove method for TdbField [R1160].....	31
Date format conversion class [R1170].....	31
Finer granularity for TdbConnectionPool synchronization [R1171].....	31
Corrected in v1.3-0.....	31
Setting a numeric field to an empty string caused exception [B1161].....	31
Exception thrown if decimal separator in locale is not a dot [B1162].....	31
Calling TdbRecord.setRecordName could throw an exception [B1168].....	31
Version 1.2-2.....	32
New Features in v1.2-2.....	32



Unicode session awareness [R1143].....	32
Corrected in v1.2-2.....	32
Exception thrown when using TdbRecordSet without a retrieval template [B1138].....	32
Version 1.2-1.....	32
New Features in v1.2-1.....	32
Connection pool shutdown.....	32
Re-establishing broken pooled sessions.....	32
Corrected in v1.2-1.....	32
Version 1.2-0.....	32
New Features in v1.2-0.....	32
Modified property access method names.....	32
Language support for connection pools.....	33
Connection pool limit and size.....	33
TRIPview 2.0 HTML highlighting.....	33
List property for assigned classification categories.....	33
Reverse retrieval of record sets.....	33
Corrected in v1.2-0.....	33
Version 1.1-0.....	33
New Features in v1.1-0.....	33
Local session support.....	33
New package name.....	33
Corrected in v1.1-0.....	33
Connection pools not working.....	33
Known Remaining Issues.....	34





## About This Document

This document contains the change history for TRIPjxp.

### Version 7.2-2

#### *New Features in v7.2-2*

##### **Check if a record has uncommitted changes [TRIP-5287]**

The TdbField and TdbRecord classes have been extended with a new method `isChanged()` that returns true if the field or record contains uncommitted changes.

#### *Corrected in v7.2-2*

##### **Record commit may not detect deleted field contents [TRIP-5280]**

Removing all values from a field would cause TRIP not to update the field upon commit if the application implemented the following sequence:

- A new TdbRecord instance is created and populated with values.
- The TdbRecord instance is committed, creating a new record in TRIP.
- Using the same TdbRecord instance, a field is cleared from all values.
- The TdbRecord instance is committed again.

### Version 7.2-1

#### *New Features in v7.2-1*

##### **Facet Baselines [TRIP-5129]**

This feature requires TRIPsystem 7.2-1 or later.

A facet baseline enables the recall of a previous, initial set of facet values so that a new facet request can 1) include the previous counts as a comparison, and 2) add values from the baseline to the new result if the new result is smaller than the requested range.

A new example program FacetBaseline has been added to exemplify the usage of this new feature. It can be found in the `com.tietoenator.trip.jxp.examples.facet` package.

##### **Database name for TdbHistoryDetail and TdbSearchSet [TRIP-5128]**

This feature requires TRIPsystem 7.2-1 or later.

The name of the database or cluster that was used to produce a search set is now available via the TdbHistoryDetail and TdbSearchSet classes.



*Corrected in v7.2-1*

### **Risk for connection failure with large facets**

When using the TdbFacet classes to generate facet values such that the total number of values for at least one facet exceeded 100, an incorrect request was sent by TRIPjxp to TRIPsystem that caused an instability that would result in a crash a couple of minutes later.

## **Version 7.2-0**

### *New Features in v7.2-0*

#### **Save part of search history as procedure [TRIP-5111]**

The TdbProcedure class has been extended with a method called saveSearch, that saves the CCL commands associated with a specified search set as a procedure, complete with dependent commands such as BASE and renumbered search set references (S=<num>).

A new example program "SearchProcedure" has been added to illustrate this new feature. It is available in the com.tietoenator.trip.jxp.examples.users package.

This feature requires TRIPsystem 7.2 or later.

#### **DB designs from TdbDatabaseClusterDesign [TRIP-5110]**

The TdbDatabaseClusterDesign class can now provide a list of all database designs for databases in a cluster, with nested clusters resolved. Use the new method getMemberDesigns() for this.

The TdbDatabaseClusterDesign.get() method can now be asked to resolve nested clusters into their constituent databases. Note that the TdbDatabaseClusterDesign instance will not support updating the cluster after this option has been used since it will represent derived information, not an actual cluster.

The new method getCommonFields() of the TdbDatabaseClusterDesign class provides access to a list of TdbFieldDesign instances that represent the fields common to all databases in the cluster.

A new example program "ShowCluster" has been added to illustrate these new features. It is available in the com.tietoenator.trip.jxp.examples.database package.

These new features require TRIPsystem 7.2 or later.

#### **New field fetch types**

New field fetch types have been added to the TdbFieldFetchType enum. When used, these new types provide additional control over what parts of a field of a record in a search set to retrieve.



## **Enhanced support for threaded applications**

Parallel requests from multiple threads are now executed in invocation order, whereas they before would be executed in random/unpredictable order depending on JVM implementation.

## **Record fetch option can distinguish between databases for the same field name**

It is now possible to set different field fetch options for the same field name in different databases. For example: the field db\_A.D as TdbFieldFetchOption.Always and db\_B.D as TdbFieldFetchOption.HitOnly.

## **HTML highlighting support for local use [JXP-107]**

Extended the HTML conversion functionality of the TdbStringField class to support hit highlighting when a local TRIPcof or TRIPviewX installation is used.

## **Support for client-side use of TRIPviewX/TRIPcof**

A local installation of TRIPviewX/TRIPcof can now be used in client-side mode when performing text extraction and HTML conversion.

*Corrected in v7.2-0*

## **TRIPjxp could hang indefinitely on server-side crash [JXP-108]**

If the tbserver process crashes or is killed, there was a risk that TRIPjxp would hang indefinitely waiting for response data from the server.

## **Client-side text extraction and HTML conversion under Linux/UNIX [JXP-110]**

Client-side text extraction and HTML conversion via TRIPviewX, TRIPview-C or TRIPcof did not work properly under non-Windows operating systems. There was a high risk that file path parameters sent to the text extraction and HTML conversion worker process would not be correctly formatted.

## **Risk for array bounds violation in client-side text extraction and HTML conversion [JXP-111]**

There was a risk for an array bounds violation in the handling of response data from the client-side text extraction and HTML conversion routines.

## **Setting part id to retrieve a single part did not work [JXP-113]**

Using the setPartId method of the TdbRecord class in order to retrieve a single part record had no effect.

## **Array bounds exception thrown if a TdbSearchSet contained a deleted record [JXP-114]**

If a record is deleted using the TdbRecord.delete() method and a search immediately thereafter is conducted via the TdbSearch class such that the resulting search set contains the record, then an array bounds exception is thrown with a message like "Array index out of range: 9" when fetching records from the set using TdbSearchSet.



This error would only occur if the database was not indexed between the delete and the search operations.

As part of the correction for this issue a new property 'deleted' has been added to the TdbRecord class. This property will return true if the object represents a deleted record.

### **Access violation exception if not logged in to session [JXP-115]**

If a TdbSession object has been created but the login() method has not been successfully called, TRIP will crash on an access violation if TRIPjxp is nevertheless being used.

TRIPjxp will now prior to any call that is not a login/authentication check the state of the session and throw a TdbException with code TdbException.NOT\_LOGGED\_IN if the login method has not been successfully called.

## **Version 3.1-0**

### *New Features in v3.1-0*

#### **Graph search and navigation [JXP-70]**

This feature requires TRIPsystem 7.1 or later.

A package com.tietoenator.trip.jxp.graph has been added. This package contains APIs for the use of graph databases in TRIP. For more information, please refer to the Programmer's Guide.

This feature requires TRIPsystem 7.1 or later.

Graph support is EXPERIMENTAL in version 7.1 of TRIPsystem and version 3.1 of TRIPjxp. Its use is encouraged and Tieto would like feedback on it. This feedback will help determine the future direction of the development of graph database support in TRIP.

#### **New method TdbSearch.addSearchSet [JXP-97]**

This new method can be used to add a pre-existing search set to a TdbSearch instance, which then assumes control over it. Once added, a TdbSearchSet instance can be obtained for the search set.

This feature requires TRIPsystem 7.1 or later.

#### **TdbSession.isConnected [JXP-106]**

A method isConnected has been added to the TdbSession class. This reports true if the session is connected to the server via a socket or if the session is web or local. False is returned for networked sessions that no longer have an active socket connection to the TRIP server.

This method is useful if TRIP access code is implemented in object finalizers that may be executed by the runtime after the session has been terminated.



### *Corrected in v3.1-0*

No defects have been corrected in version 3.1-0.

## **Version 3.0-8**

### *New Features in v3.0-8*

#### **Copy method for TdbRecord [JXP-105]**

A new method "copy" for record copying has been added to the TdbRecord class. It is strongly recommended that all existing record copying routines in TRIPjxp applications are modified to use this new method if possible. This is because it will offer better performance, plus that it fully supports STRING fields, values of which may silently be omitted from the copy when using TRIPjxp version 2.1-3 or later.

This functionality requires TRIPsystem 6.2-11 or later or TRIPsystem 7.0-4 or later in order to work.

### *Corrected in v3.0-8*

#### **Committing records from a TdbLocalSession fails [JXP-103]**

Committing a new or modified record in a TRIPjxp application that uses a TdbLocalSession to connect to TRIP would throw a NullPointerException.

#### **Using XML special characters in names may cause exception [JXP-104]**

Using XML special characters in record name and part record values could cause a TdbException to be thrown. E.g. when a quotation mark was used in the name of a record being inserted, updated or retrieved.

## **Version 3.0-7**

### *New Features in v3.0-7*

No new features have been added in v3.0-7.

### *Corrected in v3.0-7*

#### **Upload of STRINGs larger than 2MB failed to DB w/o record names [JXP-102]**

Uploading a value larger than 2MB to a STRING field in a database would fail if either 1) the STRING field is in a part record and there is no part name field, or 2) the STRING field is in the head record and there is no record name field.

This issue was also fixed in the 2.1 release series in TRIPjxp version 2.1-11.



## Version 3.0-6

### *New Features in v3.0-6*

#### **Support for pessimistic record locking [JXP-100]**

Methods lock() and unlock() have been added to the TdbRecord class in order to support pessimistic (exclusivie) record locking. This kind of locking differs from the default optimistic locking in that it blocks other user's attempts to update the record until the lock is released.

This feature requires TRIPsystem 7.0-3 or later.

### *Corrected in v3.0-6*

#### **INI-file argument to TdbTripNetSession constructor ineffective [JXP-99]**

The TdbTripNetSession constructor with four arguments takes as its third argument the name of a server-side session initialization (INI) file. The value passed to this argument did not get transferred correctly to the server, which led to the INI-file being ignored.

#### **Initialization of local TRIPcof use fails [JXP-101]**

TRIPjxp attempted to load the TRIPcof file filter configuration file using an invalid path. This resulted in TRIPcof being impossible to use locally from TRIPjxp.

## Version 3.0-5

### *New Features in v3.0-5*

No new features have been added in v3.0-5.

### *Corrected in v3.0-5*

#### **Client-side use of TRIPcof/TRIPview-C not working under Unix [JXP-96]**

Use of a local installation of TRIPview-C or TRIPcof to perform text extraction or HTML conversion did not work correctly on Linux and Unix systems. Use of a server-side installation of TRIPview, TRIPview-C or TRIPcof was unaffected by this issue.

#### **Use of TdbPoolInitializer could cause crash [JXP-98]**

There was a risk for a TdbException to be thrown when the TdbPoolInitializer was used for the execution of TRIP statements that will apply to the entire lifetime of a pooled session.

This issue was also fixed in the 2.1 release series in TRIPjxp version 2.1-10.



## Version 3.0-4

### *New Features in v3.0-4*

#### **Activity logging for connection pools [JXP-93]**

Activity logging is now supported in the TdbConnectionPool class. Assign a TdbActivityLogger instance to a pool using the new setActivityLog method. This will log the activity of the pool itself for problem analysis purposes.

This feature was also added to the 2.1 release series in TRIPjxp version 2.1-9.

#### **Overloaded acquire() method with timeout / blocking behavior [JXP-94]**

The TdbConnectionPool class has been given a new overloaded version of the acquire() method. It takes a long integer as argument that denotes a period in milliseconds to wait until a session becomes available if one is not available immediately.

This feature was also added to the 2.1 release series in TRIPjxp version 2.1-9.

#### **Session pool initialization [JXP-95]**

A new interface TdbPoolInitializer has been added to TRIPjxp. When a class implementing this interface is assigned to the connection pool right after it has been created, the pool initialization method will be called for each session added to the pool.

The use of this new interface is strongly recommended in high-load situations when several operations need to be done (e.g. opening databases, defining views, executing defines, etc) prior to the actual operation the application requires.

This feature was also added to the 2.1 release series in TRIPjxp version 2.1-9.

### *Corrected in v3.0-4*

#### **Risk for NullPointerException in TdbSession.licensedProducts() [JXP-92]**

If the TRIP license contains information about a (new) TRIP product that TRIPjxp does not know about, the licensedProducts() method would throw a NullPointerException when called.

This issue was also fixed in the 2.1 release series in TRIPjxp version 2.1-9.

## Version 3.0-3

### *New Features in v3.0-3*

#### **Support the TRIPview-C product [JXP-85]**

TRIPjxp can now detect the TRIPview-C product in the license of the currently connected TRIPsystem installation. The TdbProduct enumeration has been modified accordingly and documentation that refer to TRIPview has been modified when it also applies to TRIPview-C.





### **NOBAF option for TdbDatabaseList [JXP-89]**

Using the TdbDatabaseList class to generate a list of databases now supports a NOBAF parameter to the constructors. Setting this to true causes the list to be produced without any access to the BAF files of the databases. This gives better performance in situations with a large number of databases and when the BAF files are stored on a storage medium with slower access times.

Using the NOBAF parameter set to true means that information that is solely stored in the BAF file will not be returned. This includes the record count, which instead will be reported as the last total number of indexed records (any unindexed records will not be reported). Neither will the time for the last database update be reported.

This feature requires TRIPsystem 7.0.1 or later, and databases must be reindexed with in order for the count of indexed records to be correctly reported.

*Corrected in v3.0-3*

### **Documentation incorrect for TdbControlObject.getManagerRights [JXP-86]**

The bitmask returned by the getManagerRights method of the TdbControlObject class was incorrectly documented. It said that bit 0 indicates file management and bit 1 indicates user management, while it actually is the other way around.

### **TdbControlObject.isFileManager returned incorrect value [JXP-87]**

The isUserManager method of the TdbControlObject class returned true only if the control object represented a user that is a user manager.

### **TdbControlObject.isUserManager returned incorrect value [JXP-88]**

The isUserManager method of the TdbControlObject class returned true only if the control object represented a user that is a file manager.

### **Data retrieval from databases with classification schemes failed [JXP-90]**

Retrieval of data from any database with a classification scheme associated with it always failed with an exception indicating an out of memory error.

## **Version 3.0-2**

*New Features in v3.0-2*

### **TdbTextExtractionInfo.ExtractFromStored property added [JXP-79]**

While it has been possible to extract text from data previously stored in a STRING field, the support for this has been depending on ambiguous use of programming interfaces and success has been context dependent.

To reduce ambiguity, the TdbTextExtractionInfo class has been extended with a new property [get|set]ExtractFromStored that must be set to true if text extraction is to be done from a value previously stored in the STRING field referred to by the BinaryCopyField property. It is still also necessary to set the FileName property, but this is to inform TRIPview/TRIPcof of the file name/type. The file itself does not have to exist, nor does it get transferred to the server.





This functionality requires TRIPsystem 6.2-9:20 or later or TRIPsystem 7.0-0:4 or later in order to work.

This feature was also added to the 2.1 release series in TRIPjxp version 2.1-8.

### **Use TdbStringField for text extraction via TextExtractionInfo [JXP-83]**

It is now possible to use the data assigned to a TdbStringField instance (via the setBlob() or copyFromFile()) in text extraction specified via TdbTextField.getTextExtractionInfo().

This works so that when the new method TdbTextExtractionInfo.setStringField() is assigned a non-null value, text extraction will be performed from any value assigned to the provided TdbStringField instance. This works like if the file data had been provided to TdbTextExtractionInfo.setStream().

When the TdbTextExtractionInfo.setStringField() property is assigned a non-null value, any value passed to TdbTextExtractionInfo.setStream() will be ignored. The TdbTextExtractionInfo.setFileName() method is still required, though.

Note that if text extraction is to be performed from a value already stored in a string field, it is better not to include the TdbStringField instance at all in the TdbRecord object used for commit. In this case, pass 'true' to the TdbTextExtractionInfo.setExtractFromStored() method instead.

This functionality requires TRIPsystem 6.2-9:20 or later or TRIPsystem 7.0-0:4 or later in order to work.

This feature was also added to the 2.1 release series in TRIPjxp version 2.1-8.

*Corrected in v3.0-2*

### **Invalid characters from text extraction not handled [JXP-77]**

Text extraction (and re-extraction when creating an HTML conversion with highlight) using TRIPview or TRIPcof may sometimes produce characters that are not valid in XML text. This will result in the text not being possible to import into TRIP.

### **Text extraction using TdbStringField may fail [JXP-78]**

Text extraction from data added to a STRING field using the ExtractionTarget property of the TdbStringField class in TRIPjxp did not work if copyFromFile was used to load data into the field and setExtractionTarget was called with 'true' before the call to copyFromFile.

The operation would fail silently without throwing an exception or otherwise indicating any error. It would appear to succeed, but no content would be extracted to the referred TEXT field.

This fix was also introduced into the 2.1 release series in version 2.1-8.

### **Using TextExtractionInfo without setting Stream may fail [JXP-80]**

Using TdbTextField.TextExtractionInfo to perform text extraction without setting the Stream property would not work properly against TRIPsystem versions older than 6.2-9:8



This problem occurs with TRIPjxp versions 2.1-4 - 2.1-7 and 3.0-0 - 3.0-1. This fix was also introduced into the 2.1 release series in version 2.1-8.

#### **Text extraction update to named part records fails [JXP-81]**

Text extraction (using the classes `TdbTextExtractInfo` and `TdbTextField`) during update of a record does not work if the database has a part record name field defined. The operation appeared to succeed (no error reported), but the contents of the TEXT and STRING fields would be incorrect. Depending on usage scenario, either of these fields would be empty or retain the old value.

This problem occurs with TRIPjxp versions 2.1-4 - 2.1-7 and 3.0-0 - 3.0-1. This fix was also introduced into the 2.1 release series in version 2.1-8.

#### **TdbComponent.setName(String) fails [JXP-82]**

Creating a part record with a part name field always failed. The assignment of the name of the new part using the `setName(String)` would fail with an exception being thrown.

This fix was also introduced into the 2.1 release series in version 2.1-8.

## **Version 3.0-1**

### *New Features in v3.0-1*

#### **Keyword and metadata analysis [JXP-69]**

TRIPjxp now supports keyword and metadata analysis of text in records using TRIPcof and TRIPnlp. Analysis is performed upon commit of a new or modified record if the `TdbNlpInfo` instance accessible via the `TdbRecord.getNlpInfo()` method has been configured.

This feature requires TRIPsystem 7.0 or later with TRIPcof installed on the server side. TRIPnlp or a custom text analysis adapter for TRIPcof is also required.

#### **Use TRIPview for HTML conversion and text extraction [JXP-73]**

The `TdbStringField.convert()` method can now be used with a local installation of TRIPview 3.0 or later.

The client-side text extraction behavior of the `TdbRecord` class via the `TdbTextExtractInfo` class can now be used with a local installation of TRIPview.

#### **New method `TdbSearchSet.delete()` [JXP-74]**

A new instance method `delete()` has been added to the `TdbSearchSet` class. This method should be used for the deletion of individual search sets created via a `TdbSearch` instance instead of explicitly issuing a DELETE CCL order.

This feature was also added to the 2.1 release series in TRIPjxp version 2.1-6.



*Corrected in v3.0-1*

### **Risk for NullPointerException in TdbStringField.convert() [JXP-75]**

The TdbStringField.convert method could throw a NullPointerException when used with a STRING field containing more than 1MB of data.

### **Fetching large STRING values from part records [JXP-76]**

Incorrect part record information is can be supplied by TRIPsystem for the retrieval of large (>2MB) string field values. This causes STRING values for part record 2..N to always be fetched from the first part record instead.

When storing a large (>2MB) STRING field value for part record nr 2..N, the STRING value for the last part record would be assigned to all part records. This was due to missing part record information in the communications between TRIPsystem and TRIPjxp for the stream transfer of large binary values.

TRIPsystem version 6.2-9:18 or later and version 7.0-0:3 or later are also required on the server side to make part record storage work correctly.

This problem occurs in TRIPjxp versions 2.1-3 - 2.1-6 and in 3.0-0. This fix was also introduced into the 2.1 release series in version 2.1-7.

## **Version 3.0-0**

### *New Features in v3.0-0*

#### **Client-side text extraction [JXP-46]**

Added support for client-side text extraction using a local TRIPcof installation. Assign the ClientSide property to true on the TdbTextExtractionInfo instance for the text field in which to store the extracted text.

Do not pass file data using the only the Stream property; either pass the name of an existing local file using the FileName property, or pass the file data using a Stream and the name of the file in the FileName property.

This feature requires a local installation of TRIPcof. For usage, please look at the new sample program CofExtract and the Programmer's Guide.

#### **Client-side HTML conversion [JXP-47]**

Added support for client-side HTML conversion using a local TRIPcof installation. Use the new method TdbStringField.Convert to perform client-side conversion operations.

This feature requires a local installation of TRIPcof. For usage, please look at the new sample program CofConvert. and the Programmer's Guide.



### **TdbControlObjectList.contains methods added [JXP-48]**

The TdbControlObjectList class has been enhanced with two new versions of the contains method. One takes a name and a type and the other a name only. These can be used for more straight-forward checking of existence, e.g. the membership of users in groups.

Also see the new example ListCheckMember.

### **Session encryption [JXP-59]**

Support for encrypted sessions has been added. To establish an encrypted session, use one of the new overloaded constructors for the TdbTripNetSession class.

This feature requires TRIPsystem 7.0 or later to benefit from. With older versions of TRIPsystem, requesting an encrypted session will result in an error.

### **On-demand term lists [JXP-60]**

Support for term lists whose values are retrieved on demand has been added to the TdbSearch and TdbCclCommand classes. Terms from such term lists are retrieved on demand and in blocks of 100 terms each, which improves retrieval performance for large term lists. This behavior can be enabled using the setUsePagedTermLists (available on both the TdbCclCommand and TdbSearch classes).

Multiple on-demand term lists can also be used concurrently. See the new example program TermLists for an illustration of this behavior.

NOTE: The interface of the TdbTermList class has changed slightly in order to accommodate this enhancement. If you built your application using a version of TRIPjxp older than 3.0, you should recompile your application in order to check if you have to make any modifications to your code prior to deploying TRIPjxp.

This feature requires TRIPsystem 7.0 or later to benefit from. With older versions of TRIPsystem, all terms from a term lists will always be returned as a result of the DISPLAY command.

### **Facet classes [JXP-61]**

A new package com.tietoenator.trip.jxp.facet has been added that contains classes for generation of facets based on regular field-based term lists, KVP term lists, record classification and collections of searches.

A new example program FacetSet is provided. It illustrates the use of all four types of facets.

This feature requires TRIPsystem 7.0 or later to benefit from.

### **Access to database file location names [JXP-63]**

A class TdbLocations in the com.tietoenator.trip.jxp.databases package has been added. This class provides access to TRIP configuration entries that map to directory paths so that the corresponding symbols can be used as location names by custom database creation and alteration routines.

This feature requires file or system manager privileges in order to use.



### **Retrieve control object for a single database [JXP-66]**

The TdbDatabaseList class has been extended with an additional constructor to make it able to return a control object for a single, named database, thesarus or cluster.

This feature requires TRIPsystem 7.0 or later for best efficiency, but will work with older versions as well.

### **Synchronous indexing of databases [JXP-71]**

The TdbDatabaseDesign class has been extended with a new overloaded index method that can be used to order incremental indexing and reindexing, both in batch mode and synchronously. Note that synchronous indexing may take a long time if there is a large number of records to index.

This feature requires TRIPsystem 7.0 or later in order to work.

### **Synchronous indexing of single records [JXP-72]**

The TdbRecord class has been extended with a boolean property IndexOnCommit (setIndexOnCommit/getIndexOnCommit). By setting this to true before committing a new or updated record, the record will be synchronously indexed as part of the commit operation.

Note that batch oriented indexing remains best practise with TRIP. If synchronous indexing of single records is utilized to a great extent with a database, the scheduling of a reindex of the database from time to time is highly recommended.

This feature requires TRIPsystem 7.0 or later in order to work.

*Corrected in v3.0-0*

### **Extracting text from file with ampersand in the filename fails [JXP-55]**

Any call from TRIPjxp to TRIPview or TRIPcof to extract text from a file with an ampersand '&' in its name would fail.

### **Database language information missing from TdbDatabaseDesign [JXP-64]**

The TdbDatabaseDesign class does not provide any information about the natural language of the text in the database, nor is it possible to define this.

### **Chinese segmentation type "Word" not exposed [JXP-65]**

The chinese segmentation types All and Max are supported by the TdbDatabaseDesign class, but the type Word is reported as None. Neither is it possible to define segmentation type Word for a chinese database.



## Version 2.1-6

### *New Features in v2.1-6*

#### **New method TdbSearchSet.delete() [JXP-74]**

A new instance method delete() has been added to the TdbSearchSet class. This method should be used for the deletion of individual search sets created via a TdbSearch instance instead of explicitly issuing a DELETE CCL order.

This feature was also added to TRIPjxp 3.0 as per the 3.0-1 release, because version 2.1-6 was released after 3.0-0.

### *Corrected in v2.1-6*

No defects have been corrected in version 2.1-6.

## Version 2.1-5

### *New Features in v2.1-5*

No new features have been added in v2.1-5.

### *Corrected in v2.1-5*

#### **TdbDatabaseDesign.getVifFile returns BAF file name [JXP-62]**

When retrieving the name of the VIF file for a database using the getVifFile() method of the TdbDatabaseDesign class, the BAF file name would be returned instead.

#### **String field values may get deleted on update [JXP-67]**

When updating a record containing a string field, the string field value may get deleted if it has not been altered by the application. This problem applies to TRIPjxp version 2.1-3 and 2.1-4 when used with TRIPsystem version 6.2-9:7 or newer.

#### **TdbDateform throws exception on partial TRIP dates [JXP-68]**

A TRIP DATE value may be specified with only its year component, as well as with only its year and month component. TRIPjxp throws an exception from the TdbDateform.formatDate(String) method when trying to parse such a date.

## Version 2.1-4

### *New Features in v2.1-4*

#### **Improve STRING field upload performance [JXP-58]**

When uploading a value to a string field or to server-side text extraction, TRIPjxp may run out of memory alternatively time out when committing the new (or new version of a) record.



The string field value upload behavior has been changed so that large values are uploaded to the server in smaller chunks, which lowers memory and CPU consumption.

This feature requires TRIPsystem 6.2-9:8 or later to benefit from. With older versions of TRIPsystem, string value upload behavior will be the same as before.

*Corrected in v2.1-4*

No defects have been corrected in version 2.1-4.

## Version 2.1-3

*New Features in v2.1-3*

### **Improve STRING field download performance [JXP-56]**

When retrieving records containing large string fields, the memory usage requirements are extremely high. Fetching a single record with 100MB of binary data have been known to cause client applications to run out of memory. The same problem can manifest also with smaller string field values when retrieving a set of several records where the total size of the binary data is high.

When used with TRIPsystem version 6.2-9:7 or later, TRIPjxp will now fetch string field values on demand by utilizing a streaming mechanism added to the XPI protocol in the aforementioned version of TRIPsystem. This will lower memory usage and reduce the overall retrieval and processing time for records and search results.

This feature requires TRIPsystem 6.2-9:7 or later to benefit from. With older versions of TRIPsystem, string values will be fetched inline like before.

*Corrected in v2.1-3*

### **TdbException.getMessage() not complete [JXP-57]**

The getMessage() method of the TdbException class would in some cases not return a stack trace. Stack traces for nested exceptions would only be one level deep.

## Version 2.1-2

*New Features in v2.1-2*

### **TdbDatabaseDesign.getOwnerName method added [JXP-50]**

The TdbDatabaseDesign class has been enhanced with a method getOwnerName that returns the name of the user who owns the database.

This feature requires TRIPsystem version 6.2-8:4 or later.





### **Filename field name for HTML conversion [JXP-53]**

There are now new constructors for the classes TdbRendition and TdbHighlightRendition that take the name of a phrase field. This phrase field must contain the original name of the file stored in the string field that is about to be converted to HTML.

This information is necessary for TRIPview so that it will better be able to identify the type of the file. If the original file name is not known to the HTML converter, there is a risk that the conversion will fail.

This feature requires TRIPsystem version 6.2-9 or later.

### **Access rights via group memberships from TdbUserAccessList [JXP-54]**

A new constructor to TdbUserAccessList has been added that takes a boolean parameter that indicates if access rights granted via group memberships also should be returned.

This feature requires TRIPsystem version 6.2-9 or later.

*Corrected in v2.1-2*

### **Invalid exception from TdbSearchSet.clearCache() [JXP-51]**

The clearCache() method in the TdbSearchSet class would incorrectly throw an exception if called before at least one record was returned to the application via the current TdbSearchSet instance.

### **No output to activity log if server exits prematurely [JXP-52]**

The TRIPjxp activity log would not produce any output for an XPI transaction if the server hangs or crashes. Neither request nor response would get logged.

## **Version 2.1-1**

*New Features in v2.1-1*

### **Support for search comforters [JXP-45]**

Support for search comforters has been added to TRIPjxp. This makes it possible to get regular notifications ("searching...") from the server for lengthy queries, and also enables applications to abort such queries.

The sample TRIPtty.java has been extended to include comforter notification support.

This feature requires TRIPsystem 6.2-8 or later.

*Corrected in v2.1-1*

No defects have been corrected in version 2.1-1.





## Version 2.1-0

### *New Features in v2.1-0*

#### **Open up for segmentation type "Word" [JXP-39]**

Enabled assignment of the Chinese text segmentation type "Word" to a database design. Existing database designs with this segmentation type will be correctly shown as "Word" instead of "None", which also is used in TRIPjxp for unknown or invalid segmentation values.

#### **XPath as Query Language [JXP-40]**

Enhancements have been made to the TdbSearch, TdbRecordSet, and TdbCclCommand classes so that they can be used to query TRIPxml databases using XPath expressions.

This feature requires TRIPsystem 6.2-5 or later and TRIPxml 3.0 or later. Please refer to the TRIPxml manual for valid XPath syntax.

#### **XML Fragment Retrieval [JXP-41]**

It is now possible to retrieve XML fragments from a search set containing XML records. The node set that form the roots of the fragments is specified by an XPath expression via the new TdbSearchSet method asFragmentSet().

This feature requires TRIPsystem 6.2-5 or later and TRIPxml 3.0 or later. Please refer to the TRIPxml manual for valid XPath syntax.

### *Corrected in v2.1-0*

#### **TdbTuple.setValue(String,String) always throws exception [JXP-43]**

Using the TdbTuple.setValue(String,String) method to assign a value to a tuple always results in an exception being thrown saying that "There is no field <1> in this tuple list."

#### **Exception when creating a record in DB with a record id field [JXP-44]**

Creating a new record in a database with a record id field may throw an exception in TRIPjxp upon completion. The record will be successfully created, however.

## Version 2.0-4

### *New Features in v2.0-4*

No new features have been added in v2.0-4.



*Corrected in v2.0-4*

#### **Changing password for another user does not work [JXP-37]**

Being logged on as SYSTEM and using the resetPassword method of the TdbUser class to change the password for another user did not work. Success would be reported, but the password had not changed into the specified new password. The old password would not work either after the resetPassword call.

#### **Changing password for current user does not work [JXP-38]**

Using the changePassword method of the TdbUser class to change change the password of the current user did not work. The request would fail with the message "invalid arguments passed to function".

## **Version 2.0-3**

*New Features in v2.0-3*

#### **Control object method for checking if a database is XML enabled [R1460]**

The TdbControlObject class has a new method isXmlEnabled(), which returns true if the database that the control object represents is XML enabled.

#### **Enable custom ASE calls for insert and update of records [R1457]**

It is now possible to assign names of ASE routines to call when updating or inserting a record. The TdbRecord class have two new methods for this: insertAseList and updateAseList. Add the names of the ASE routines to call to these lists, and they get called in order.

There is a new section at the end of chapter 7 in the TRIPnpx & TRIPjxp Programmer's Guide that explains this feature.

This behavior requires TRIPsystem 6.2-4 or later.

*Corrected in v2.0-3*

No defects have been corrected in version 2.0-3.

## **Version 2.0-2**

*New Features in v2.0-2*

#### **Automatically close idle sessions in connection pool [R1415]**

It is now possible to have a connection pool automatically close sessions that have been idle for more than a specified number of seconds.

New properties have been added to the TdbConnectionPool class to accomplish this; purgeInterval, idleTimeout and minIdle.

*Corrected in v2.0-2*

No defects have been corrected in version 2.0-2.



## Version 2.0-1

### *New Features in v2.0-1*

#### **Finer control when specifying what to fetch from a TEXT field [R1411]**

The TdbFieldTemplate class has changed so that the 'focus' property now is split into three different modes. 'Focus' means to only fetch a keyword-in-context representation of a field with hits. If no hits are available, nothing will be returned. 'Extract' means to extract a specified number of characters from the start of the field. 'FocusOrExtract' first attempts to do a keyword-in-context, but if the field contains no hits, then the specified first number of characters from the field will be returned instead. This corresponds to the behavior in version 2.0-0 and earlier.

All the new fetch modes for a field is available via the new enum TdbFieldFetchType. The TdbFieldTemplate properties Focus and Summary are marked as obsolete and will be removed in the next major release.

Proper use of the TdbFieldTemplate class requires TRIPsystem version 6.2-2 or later. Earlier versions of TRIPsystem will also work, but the fetch types 'Extract' and 'Focus' will not work.

### *Corrected in v2.0-1*

No defects have been corrected in version 2.0-1.

## Version 2.0-0

### *New Features in v2.0-0*

#### **Method TdbDatabaseClusterDesign.remove() added [R1404]**

A remove() method has been added to the TdbDatabaseClusterDesign class in order to make removal of cluster members easier.

#### **Get field value size [R1372]**

The total size of the value in a field can now be accessed via the TdbField method getFieldSize().

If the application only needs the size of the field and not the value itself, note that a significant overhead can be incurred. In order to avoid that, there is a new method setFieldContent(boolean) on the TdbFieldTemplate class that, if passed 'false', disables retrieval of the field value while retaining the capability to return the size of the field value to the application.

This behavior requires TRIPsystem 6.2-1 or later.

#### **Access field template collection via TdbRecord [R1346]**

The class TdbRecord contains a new method "getFieldTemplates", that returns an Iterator<TdbFieldTemplate> object that can be used to inspect the field templates in a retrieval template.



### **FieldCount property for TdbDatabaseDesign [R1345]**

The new TdbDatabaseDesign method getFieldCount provides a count of the number of fields in the database design.

### **New class: TdbSearch [R1344]**

The new class TdbSearch provides a search interface that combines the best of the classes in the ccl and data packages with features gathered from best practices and observations regarding ease of use of search capabilities in earlier versions of TRIPjxp.

The TdbSearch class is demonstrated in the new examples "Search" and "SearchSetIteration".

The TdbSearch class is EXPERIMENTAL in this release.

### **Support for thesaurus-based searching in TdbRecordSet [R1333]**

Via the new method setThesaurus on the TdbRecordSet object, it is now possible to define a thesaurus to use with the CCL order. This makes it possible to use thesaurus-specific syntax, e.g. DOWN() and UP().

### **Access record timestamp through TdbRecord [R1316]**

Via the new method getTimeStamp() on the TdbRecord object, it is now possible to get a Java Date version of the timestamp for stored records.

### **Record ID available for new records [R1305]**

When creating a new record using the TdbRecord or TdbXmlRecord classes, the method getRecordId() returns the record ID of the new record after successful completion of the commit() call.

### **TdbDateform capable of formatting TRIP time stamps [R1300]**

It is now possible to use the TdbDateform class to format TRIP time stamp values such as returned by the LastUpdate property of the TdbDatabaseDesign class.

### **Session isolation for connection pools [R1291]**

It is now possible to control the session isolation behavior for connection pools. By default, sessions in pools are isolated from each other, but this may now be disabled, so that the automatic SAVE/LOAD STATUS that otherwise generate network I/O is not performed.

### **Support for TRIPxml 2.x added [R1124]**

Via the new class TdbXmlRecord, it is now possible to import and export documents to/from a TRIPxml database. This feature requires TRIPsystem version 6.2-1 or later.

*Corrected in v2.0-0*

### **Exception thrown when saving database cluster design changes [B1403]**

Modifying a cluster via an instance of the TdbDatabaseClusterDesign class initialized via a call to get(string) would cause an exception to be thrown when calling the put() method to commit the changes.



## **Base64 routines produce invalid encoding [B1294]**

The Base64 encoding is used to transmit STRING field values and output format results via the XPI protocol.

Unfortunately, the data produced by the the encoder and decoder was not valid according to the RFC 4648. In order to fix this problem, the server-side base64 routines has been modified. The client-side decoder in TRIPjxp has consequently also been modified to reflect the necessary changes.

One of the noticeable symptoms of this were that output data could sometimes be a couple of bytes larger than what was originally sent in.

A side effect of this bug fix is that when connecting to a TRIPsystem installation older than version 6.2, the login attempt will at first fail even if a correct password was provided. TRIPjxp will then automatically fall back to the old version of the Base64 encoding and retry the login. The user will not notice anything, but the two separate login attempts can be detected on the server if logging is enabled.

## **Version 1.3-4**

### *New Features in v1.3-4*

#### **Access to the did-you-mean facility [R1256]**

In preparation for the release of TRIPsystem 6.2, the classes TdbCclCommand and TdbRecordSet have been enhanced with the property getter method `getDidYouMean()`. When used with TRIPsystem 6.2 and later, any FUZZ command issued can result in a suggestion for an alternative query that is likely to give a greater amount of hits.

### *Corrected in v1.3-4*

#### **Less-than sign in command text caused corrupt XPI request [B1250]**

A TdbException with the message "illegal order part" was thrown when searches were done via the TdbCclCommand class using the lower-than-or-equal operator, e.g. "FIND FLD<=2009".

When searching via the TdbCclCommand class using the lower-than operator (e.g. "FIND FLD<2009") invalid hits or no hits at all would be returned.

## **Version 1.3-3**

### *New Features in v1.3-3*

No new features have been added in v1.3-3.

### *Corrected in v1.3-3*

#### **Months incorrectly parsed in TdbDateform.formatTimestamp [B1221]**

The month in the timestamp string produced by the TdbDateform.formatTimestamp method is incorrect. The method produces April for March, November for October, etc.



### **Risk for exception in TdbDatabaseAccess.get for empty scopes [B1244]**

The server may sometimes send an XPI response with an empty node for a read or write scope where TRIPjxp assumed the existence of a text node for the scope definition. This special case was incorrectly handled in TRIPjxp when parsing the response.

## **Version 1.3-2**

### *New Features in v1.3-2*

#### **TRIPgrid added to TdbProduct enum [R1199]**

TRIPgrid has been added to the TdbProduct enumeration, which makes it possible to check if the license on the currently connected server includes TRIPgrid.

### *Corrected in v1.3-2*

#### **Record name not parsed for databases not owned [B1214]**

The structure of the BASE\_RES message from the server when calling the TdbDatabaseDesign methods get() or get(name) is different depending on whether the logged on user is the owner of the database or not.

TRIPjxp did not pick up the record name correctly if the logged on user is not the owner of the requested database design.

## **Version 1.3-1**

### *New Features in v1.3-1*

No new features have been added in v1.3-1.

### *Corrected in v1.3-1*

#### **Removal of all values from a field not working [B1182]**

If all values in a structured field were removed (e.g. via the TdbField.clear() method), no information about the removal was sent to the server upon commit. This resulted in that the field was totally unaffected, and the values remained in the field. This applies to structured fields only.

## **Version 1.3-0**

### *New Features in v1.3-0*

#### **Tuple lists [R1152]**

Tuple lists are now supported via the two new classes TdbTupleList and TdbTuple for all fields that are capable of having subfields, i.e. fields of type Phrase, Integer, Number, Date and Time.



There is a new section 8 in the Programmer's Guide that describes how to use tuple lists. There is also a new example program named "Tuples".

### **New method TdbField.getValueCount [R1158]**

The method TdbField.getValueCount has been added that reports the number of values (subfields) in the current field.

Previously, the only way to get a subfield count was to access the list returned by the values() method. This was not optimal, since that collection is re-created every time it is accessed.

### **Value insert method for TdbField [R1159]**

The method TdbField.insertValue has been added that enables the insertion of a value in the middle of the value list (subfields).

### **Value remove method for TdbField [R1160]**

The method TdbField.removeValue has been added that enables the removal of a single value from a field. The only way to accomplish this prior to release 1.3-0 was to clear all values from the field, then adding the required ones back.

### **Date format conversion class [R1170]**

There is a new class called TdbDateform that offers methods for conversion of Java Date objects to and from the current TRIP date format. It is accessible via the method getCurrentDateform() on the TdbSession class.

### **Finer granularity for TdbConnectionPool synchronization [R1171]**

The synchronization in the TdbConnectionPool class is now more fine grained, which allows more instructions to be run without any need for synchronization.

*Corrected in v1.3-0*

### **Setting a numeric field to an empty string caused exception [B1161]**

Passing an empty string to a numeric field caused TRIPjxp to throw an exception upon format validation.

### **Exception thrown if decimal separator in locale is not a dot [B1162]**

The TRIPxpi server-side library does not accept any other decimal separator than a dot. A locale that specifies something else, e.g. a comma, caused a formatting error.

Field values passed to and obtained from the TdbNumberField class now uses the decimal separator as defined in the current locale. Any required conversion is done internally by TRIPjxp.

### **Calling TdbRecord.setRecordName could throw an exception [B1168]**

Calling TdbRecord.setRecordName on a new record would throw an exception if the head component contained a TdbField object for the record name field and no value previously existed.





## Version 1.2-2

### *New Features in v1.2-2*

#### **Unicode session awareness [R1143]**

As per TRIPsystem 6.0-2, the TRIPxpi always starts sessions in unicode mode. The methods `TdbSession.getCharacterSet()` and `TdbSession.getCharacterSetName()` are now able to return the unicode character set enumerated type and name instead of incorrectly reporting ISOLatin1/LA1 as the fallback for an unknown character set.

### *Corrected in v1.2-2*

#### **Exception thrown when using `TdbRecordSet` without a retrieval template [B1138]**

Using the `TdbRecordSet` without a retrieval template as documented on page number 39 in the Programmer's Guide threw a `NullPointerException` instead of reporting the number of records found.

## Version 1.2-1

### *New Features in v1.2-1*

#### **Connection pool shutdown**

A method `close()` has been added to the `TdbConnectionPool` class. Call it to cleanly shut down the pool on application exit.

#### **Re-establishing broken pooled sessions**

The `TdbConnectionPool` class will now attempt to re-establish broken sessions (e.g. ones that have lost connection to the server).

Two new properties, `RestoreRetries` and `RestoreDelay`, have been added to support this behavior. The `RestoreRetries` property represents the max number of retries to use for re-establishing a broken session. The `RestoreDelay` property represents the number of milliseconds to wait between session attempts to re-establish a broken session.

### *Corrected in v1.2-1*

No defects have been corrected in version 1.2-1.

## Version 1.2-0

### *New Features in v1.2-0*

#### **Modified property access method names**

Property accessors (`get/set`) are now named using the JavaBean naming conventions wherever possible.





## Language support for connection pools

A connection pool can now be associated with a CCL dialect.

## Connection pool limit and size

Two new properties on the TdbConnectionPool class (Limit and Size) allows for control of the number of sessions that can be maintained by the connection pool.

## TRIPview 2.0 HTML highlighting

The class TdbHighlightRendition was added to support the retrieval of highlighted HTML conversions of STRING field contents. Use of this feature requires TRIPview 2.0 on the server.

## List property for assigned classification categories

Via the method getCategories(), the TdbRecord class can return a list of assigned classification categories for the record. This list will be empty unless the database is configured with a classification scheme.

## Reverse retrieval of record sets

Setting the property RetrieveReversed on a TdbRecordSet instance can be used to retrieve the requested set of records in reverse order. This is equivalent of the CCL command "SHOW REVERSE".

## *Corrected in v1.2-0*

No defects have been corrected in version 1.2-0.

# Version 1.1-0

## *New Features in v1.1-0*

## Local session support

TRIPjxp now supports local sessions via the TdbLocalSession class. Local sessions connect to TRIPsystem running on the local machine via direct JNI calls to the TRIP kernel. In order for this to work, the following requirements must be met:

- The JAR file "tripjxp\_local.jar" must be on the CLASSPATH.
- The JNI library jxplocal (i.e. "JXPLOCAL.DLL" on, or libjxplocal.so on Linux/Unix) must be on the library path, or specified using the system property java.library.path.

## New package name

The word 'trip' has been inserted into the package name. The jxp packages are now all prefixed by com.tietoenator.trip.jxp.

## *Corrected in v1.1-0*

## Connection pools not working

Trying to create a connection pool resulted in a TdbException with the TdbException.UNUSABLE\_SESSION code.



## Known Remaining Issues

No issues are currently known.