



SpeedDBee

— KVS 版の Index 機能 —

Version 6.0

ユーザガイド

株式会社ソルティスター

● 変更履歴

DOC ver	KVS ver	日付	変更内容
V6.0.0	V6.0.0	2020/06/11	新規作成

目次

1. はじめに	4
2. SpeeDBee/KVS インデックス概要	4
3. 全体構成と管理	4
4. 機能	6
5. データ型	9
6. 文字コード	10
7. 制限事項	10
8. API 仕様	11
8.1. 基本データ型	11
8.2. インターフェース初期化/終了	11
8.3. ワークスペース操作	12
8.4. インデックス定義	16
8.5. インデックス更新	17
8.6. トランザクション	19
8.7. 外部リストインデックス	20
8.8. インデックス検索	21
8.9. インデックス学習補助機能	24
8.10. メンテナンス	24
9. 暗号化	25
9.1. 暗号化の対象	25
10. ユーザー定義データ型	26
10.1. ユーザー定義データ型の定義	26
10.2. ユーザー定義データ型のインデックス作成	26
10.3. ユーザー定義データ型の関数テーブルへの登録	26
11. リングストレージ機能	27
11.1. リングストレージ定義の指定	27
11.2. 最大サイズ指定 EGI_IDX_OPT_AUTO_DEL_WKS	28
11.3. 最大件数指定 EGI_IDX_OPT_AUTO_DEL_IDX	28

1.はじめに

本書は SpeedBee/KVS の Index 機能を説明します。SpeedBee/KVS では、B-Tree（完全一致/前方一致/範囲）、N-GRAM（フリーワード/曖昧）、Z-Order（空間）、K-Tree（絞り込み）の各種データ検索用のインデックスを提供します。本書では、各種インデックスの概要、使い方を説明します。詳細は、API リファレンスを参照願います。

2.SpeedBee/KVS インデックス概要

インデックスは索引キーとなるデータを基に該当データを特定するための仕組みです。予めキーデータと ID データを学習させておくことで検索対象データを高速に特定します。SpeedBee/KVS では用途に応じた多種多様なインデックスを提供します。インターフェースとして、eNoS-API を利用することで今まで個々に開発していた検索アルゴリズムを統合し、最適なパフォーマンス、高い生産性を得られます。以降、検索キーをキーデータ、検索対象データに相当する識別子(データを含む)を ID データとします。

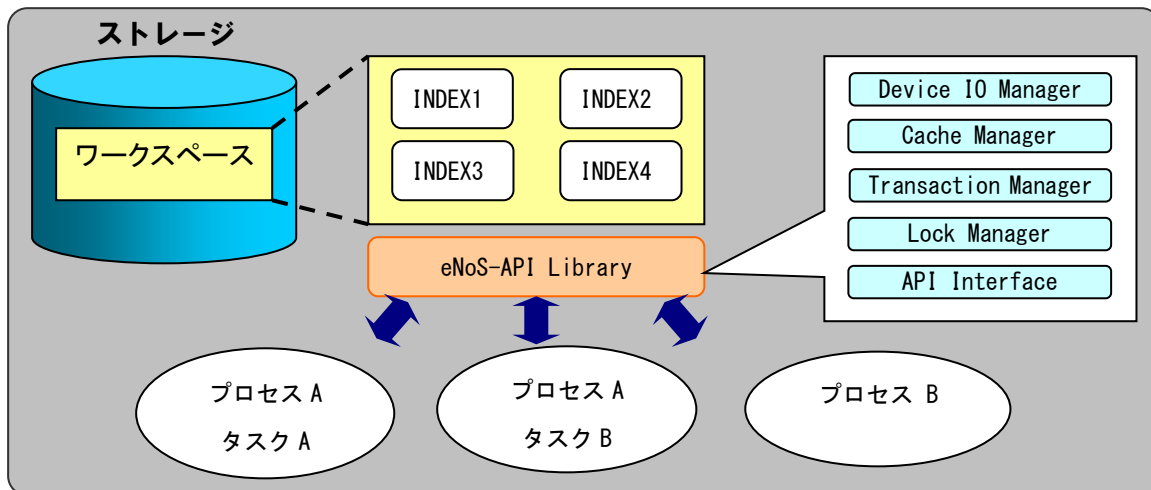
API 定義データ型

#	API 定義のデータ型	C データ型
1	B	char
	UB	unsigned char
	H	short
	UH	unsigned short
	W	int
	UW	unsigned int
	VP	void *
	INT	int
	UINT	unsigned int

3.全体構成と管理

SpeedBee/KVS はワークスペースと呼ばれる領域を永続媒体（メモリや HDD）上に作成します。ワークスペース上に用途に応じたアルゴリズム(種類)でインデックスを複数作成することができます。

作成されたインデックスは、アルゴリズムに関係なく、共通のインターフェース (eNoS-API) にて操作します。

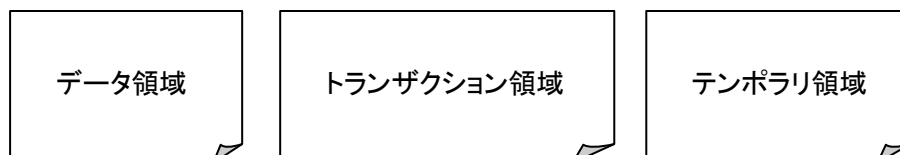


SpeedBee/KVS で管理されるワークスペースには I/O デバイスに依存した幾つかのストレージタイプがあり、タイプにより管理形態が変わります。

ストレージ	管理形態	概要
FILE	分割	一般的なファイルシステム上に作成される SpeedBee/KVS ワークスペースで I/O ドライバが別途必要になります。このタイプでは、管理形態を分割または単一管理のどちらかを指定することができます。
MEMORY	分割	揮発性メモリ上に作成される SpeedBee/KVS ワークスペース（オプション）
DEVICE	単一	ファイルシステムを持たない物理デバイス上に作成される SpeedBee/KVS ワークスペースで I/O ドライバが別途必要になります。（オプション）
SQLITE	分割	SQLITE 連携で使用される SpeedBee/KVS ワークスペースで、SQLite データベース内にワークスペースを作成します。SQLite 環境が必要であるため、このストレージタイプの I/O ドライバはオプションであり、標準では付属していません。
NETWORK	単一	本バージョンではサポートしていません。

✓ 分割管理

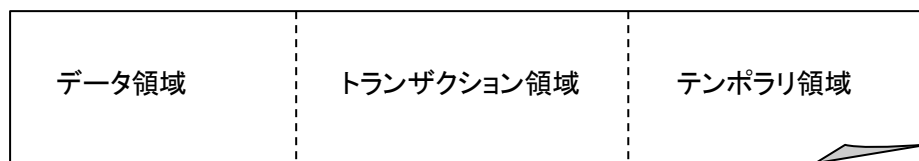
SpeedBee/KVS のワークスペースは、データ領域／トランザクション領域／テンポラリ領域の3つの領域に分けられます。分割管理では領域をそれぞれに管理し、各領域は可変長領域となります。



✓ 単一管理

分割管理はデータ領域／トランザクション領域／テンポラリ領域を連続した1つの領域で管理します。

予め各領域のサイズを指定し、その範囲内で各領域を管理します。予め割り当てられた領域以上に管理領域を拡張することはありません。



※作成されたトランザクション／テンポラリ領域は不要になった際も削除されません。

※領域の順番は図のようにデータ領域、トランザクション領域、テンポラリ領域の順になります。

4.機能

SpeedBee/KVS の Index 機能では下記の機能を提供します。

✓ ワークスペースの管理

インデックスを作成するワークスペース領域を管理する機能を提供します。

ワークスペース領域は作成時における設定で利用量を制限することが可能です。

✓ インデックスの作成・削除

インデックスを作成・削除する機能を提供します。インデックス作成はキーデータのデータ型を指定し、システム要件に適切なインデックス方式を選択することで行います。

○ インデックスに格納できるデータ型について

整数型、浮動小数型、文字列型、時間型、座標型のデータを格納できます。

○ ID データについて

インデックスを構築する際、キーデータと ID データが必要になります。ID データはアプリケーション側にて、データを特定するための ID として定義する必要があります。ID データの格納サイズは、インデックス構築時に指定します。

○ 作成できるインデックス方式について

M-TREE	...	可変長 ID データを格納し、特定のキーデータからその可変長 ID データを取得することができるインデックスです。B-TREE アルゴリズムで管理され、完全一致、前方一致、範囲条件で検索することが可能です。登録するキーデータはユニークである必要があります。
B-TREE	...	B-TREE (Balance Tree) を利用したインデックスです。RDBMS など一般的によく利用されます。完全一致、前方一致、範囲条件で検索することが可能です。
N-GRAM	...	文字列を任意の文字数で分割しインデックスを構築します。文字列型に対する部分一致検索が可能です。
Z-ORDER	...	2次元座標を合成し、Z 矩形の軌跡をたどる形態でインデックスを構築します。座標型のキーデータを格納することができ、完全一致、2次元範囲で検索することが可能です。

K-TREE	...	よみ仮名(半角カタカナ)を1文字単位でTREE管理するインデックスを構築します。 文字列型で半角カタカナを格納でき、1文字単位でのインクリメントサーチ(*1)が可能です。 入力文字単位にて、該当件数、次候補文字を高速に取得することができる特殊なインデックスです。 *1 一文字入力するたびに一致する候補を絞り込む検索。
--------	-----	--

✓ データの更新(登録・削除・更新)

作成したインデックスに対するデータ登録、削除、更新が行えます。

更新は M-TREE インデックスの ID データに対する更新となり、その他インデックスでは登録、削除のみが可能です。

✓ トランザクション

トランザクション機能をサポートし、データ更新時の整合性を保証します。

データ更新操作およびインデックス DDL 操作では、暗黙的にオートコミット機能(自動的なトランザクション操作)が働きます。また、オートコミット機能を使わずに明示的にトランザクションの開始(BEGIN) および終了(COMMIT, ROLLBACK)を指示して操作することもできます。トランザクション分離レベルは READ COMMITTED でコミット済みのデータのみ読み取ることができます。

トランザクションリソースはワークスペースに対し1つであり、同じワークスペースに対して複数のプロセスおよびスレッドから同時にトランザクションをかけることはできません。

✓ ロック

マルチプロセスおよびマルチスレッドによるデータ操作においてデータ整合性を保つために内部的にロック機構が働きます。

ロックには、表面的には(※1) DB ロックとリードロックがあります。

DB ロックは、ワークスペースに対しての排他ロックです。このロックがかけられた場合、他のプロセスおよびスレッドからはワークスペースへのアクセスは一切できません。DB ロックは、以下の操作が実行されたときに働きます。

- ・ トランザクションコミット時のデータ領域への更新の間
- ・ インデックス DDL 操作の間

リードロックは、ワークスペースの参照操作をしている場合にかかけられます。リードロックがかけられている間は、DB ロックをかけることはできません。

- ・ 検索処理プリペアカーソル処理からカーソルクローズまで

※1 内部的には他のロックの種類があります。

✓ データの検索

下記のインデックス検索方式を提供します。

全件検索	...	インデックスに登録されている全件データ
完全一致	...	値が等しいデータ
前方一致	...	~で始まるデータ
範囲検索	...	最低値 $\leq X \leq$ 最大値 に該当するデータ
部分一致	...	文字列の一部に指定キーワードが存在するデータ
座標範囲	...	$(X1, Y1) \leq (X, Y) \leq (X2, Y2)$ に該当するデータ
完全一致 (複数)	...	$X = 'X1' \text{ or } 'X2' \text{ or } 'X3'$ に該当するデータ

✓ ソート

取得 ID データのソート順はインデックスアルゴリズムに依存します。

M-TREE	...	登録キーデータのソート順で ID データが取得されます。
B-TREE	...	登録キーデータのソート順で ID データが取得されます。
N-GRAM	...	ID データのソート順で ID データが取得されます。
Z-ORDER	...	登録キーデータのソート順で ID データが取得されます。
K-TREE	...	ソート順は保証されません。

✓ 暗号化

ワークスペースに含まれるデータの暗号化に対応します。暗号化アルゴリズムはユーザードライバ内で任意のアルゴリズムを組み込む必要があります。暗号化の単位としてはブロック暗号を基本とします。(詳細は「10. 暗号化」を参照)

✓ データファイルの互換性

SpeedBee/KVS のデータファイルのバイトオーダーは、どのようなマシン環境においても互換性をもたせるためにポータリング時にエンディアンを指定することができ、ハードウェアに依存しないフォーマットで作成されます。

✓ リングストレージ機能

リングストレージ機能は、限られたストレージリソースにおいて指定された最大サイズまたは最大件数を越えた場合、自動的に古いデータを削除し、登録スペースを空けることでメンテナンスなしに永続的に登録し続けられる機能です。

この機能は MTREE および BTREE (UNIQUE) インデックスではリングストレージ機能を利用することができます。

✓ メモリ管理機能

環境パラメータ ECI_MEM_LIMIT_SIZE を指定した場合、ライブラリ内部のメモリアロケータを使用し、ライブラリ内で使用するメモリ利用サイズの制限およびメモリ使用情報を取得することができます。(詳細は「8.12 メモリ利用情報取得 API」を参照)

✓ ワークスペース認証機能

ワークスペースファイルを他のマシンへコピーした場合、データアクセスが可能でセキュリティ上問題があるため、ワークスペースを作成したマシンのみアクセス可能でかつコピー先ではアクセスできないようにするための認証機能を提供します。

5. データ型

SpeedBee/KVS にて定義されるデータ型とその表現範囲を記します。

データ型	サイズ	範囲	
int8	1	-128	~ 127
int16	2	-32768	~ 32767
int32	4	-2147483648	~ 2147483647
int64	8	-9, 223, 372, 036, 854, 775, 808	9, 223, 372, 036, 854, 775, 807
uint8	1	0	~ 255
uint16	2	0	~ 65535
uint32	4	0	~ 4294967295
uint64	8	0	18, 446, 744, 073, 709, 551, 6157
float	4	-3. 402823466e+38F	~ 3. 402823466e+38F
double	8	-1. 7976931348623158e+308	~ 1. 7976931348623158e+308
varchar	ユーザーにてサイズを指定します。最大 512 文字までの指定が可能です。		
text	サイズ指定はありません。		
date	4	0000/01/01	~ 9999/12/31
time ※1	8	1970/01/01 00:00:00	~ 9999/12/31 23:59:59
timestamp ※1	12	1970/01/01 00:00:00 000000	~ 9999/12/31 23:59:59 999999
point	8	(0, 0)	~ (4294967295, 4294967295)
auto-number ※2	4	1	~ 4294967295
auto-number64 ※2	8	1	18, 446, 744, 073, 709, 551, 6157
user define data type (UDD) ※3	ユーザー定義型		

※1 TIME 型、TIMESTAMP 型

TIME 型、TIMESTAMP 型は、入出力はローカルタイムとして扱い、データは GMT に変換され格納されます。ローカルタイムはシステムおよびユーザー環境のタイムゾーンに従います。(ただし SpeedBee/KVS 環境パラメータ ECL_GMT_TIME が設定されている場合は、入出力も GMT のまま扱います。)

※2 AUTO-NUMBER 型 AUTO-NUMBER64 型

M-TREE インデックスのキーデータのデータ型として使用できます。それ以外では使用できません。auto-number 型は登録 ID データに対するキーデータの値をインデックス内のユニークな整数値として決定します。auto-number 型で決まる整数値はインデックス内における最大値+1 になります。auto-number 型の最大値を超えた場合、再度 1 から始まります。M-TREE はユニーク制約があるため数値が重複した場合、登録エラーになるので注意してください。

※3 USER DEFINE DATA TYPE(UDD) ユーザー定義データ型

ユーザー指定のデータ型を扱うことが可能でM-TREEおよびBTREEインデックスのキーデータのデータ型として利用できます。利用する場合はユーザードライバ内にユーザー定義データ型の比較関数、コピー関数を定義する必要があります。(詳しくは 11.ユーザー定義データ型を参照)

6.文字コード

文字コードはワークスペース作成時に決定し、インデックス全ての文字列処理は指定された文字コードで処理されます。指定可能な文字コードは次になります。

デフォルトの文字コードは SJIS で、ワークスペースパラメータの LANG で設定ができます。

- ・ SJIS (Shift-JIS)
- ・ EUC (EUC-JP)
- ・ UTF8
- ・ UTF-16LE
- ・ UTF-16BE

7.制限事項

- ワークスペース内に作成できるインデックスは最大 256 個です。
- 通常、ID データは符号なし整数で 16/32/64 ビットの何れかを指定することになります。64 ビット整数型を持たないプラットフォームでは 16 または 32 ビット整数のみとなります。M-TREE では登録する可変長 ID データサイズの上限が 65535Byte になります。必ず 1Byte 以上の ID データである必要があります。
- 通常、登録可能な文字列型のキーデータはバイトサイズ制限なしの最大 512 文字になります。文字列型以外は各データ型のサイズに依存します。K-TREE インデックスでの登録可能な文字数は 127 文字となり、1 バイトで表現できる文字が対象となります。N-GRAM インデックスに関してはサイズ制限がありません。
- 各インデックスにおける制限事項一覧を下記に記します。

インデックス	キーデータのデータ型	キーデータ最大文字数 (文字)	ID データサイズ (byte)	文字コード
M-TREE	TEXT 型を除く 全てのデータ型	512	1~65535	-
B-TREE	AUTO-NUMBER、TEXT 型 を除く全てのデータ型	512	2, 4, 8	-
N-GRAM	TEXT 型	-	2, 4, 8	-
Z-ORDER	座標型	-	2, 4, 8	×
K-TREE	VARCHAR 型	127	2, 4, 8	SJIS

※その他、API によって制限事項がありますが、各 API 仕様の説明を参照してください。

8.API 仕様

8.1. 基本データ型

API で使用される基本データ型は次になります。

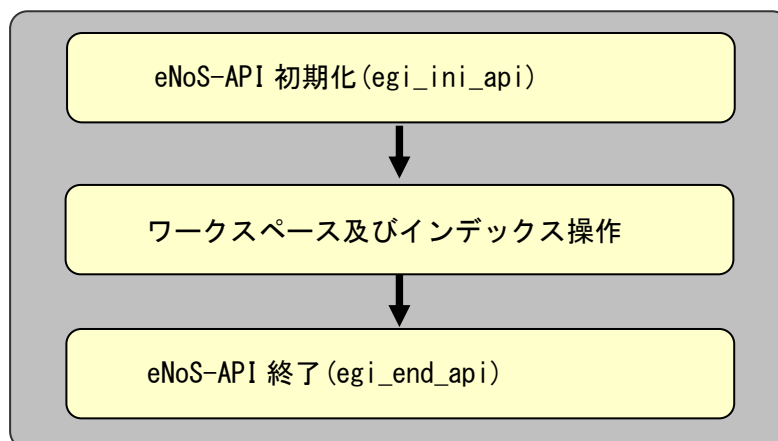
型	C 言語タイプ	備考
B	char	
UB	unsigned char	
H	short	
UH	unsigned short	
W	int	
UW	unsigned int	
D	long long	__int64
UD	unsigned long long	unsigned __int64
VP	void*	
BEE_RET	int	eNoS-API の戻り値として使用されます。
BEE_ERR	int	エラーコード値として使用されます。
BEE_MB	unsigned char	マルチバイト文字を扱う際に使用されます。*1
BEE_WC	unsigned short	ワイド文字を扱う際に使用されます。*2

*1 SJIS、EUC、UTF-8 を扱います。

*2 UTF-16LE UTF-16BE を扱います。

8.2. インターフェース初期化/終了

SpeedBee/KVS-API を使用するにあたり、アプリケーション開始/終了時にインターフェースの初期化及び終了処理を行う必要があります。



8.3. ワークスペース操作

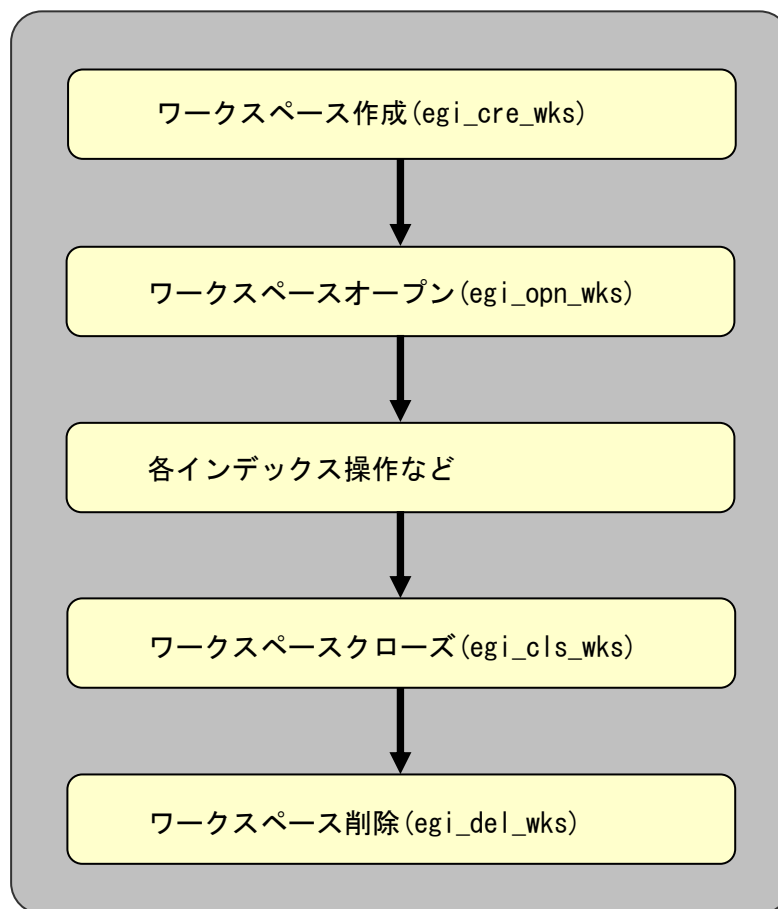
●ワークスペース基本操作

ワークスペースはインデックス管理情報とインデックスデータをグループ化したものでインデックス作成前に作成しておく必要があります。また、既に存在するワークスペースに対して操作することも可能です。

ワークスペース 1 回のオープンで各インデックス操作が可能となり、インデックス操作毎に毎回オープンする必要はありません。

ワークスペースの基本操作は次のようになります。

下記フローでワークスペース作成に関しては初回起動時に必要となり、削除に関しては必要に応じて実行することになります。



●ワークスペースパラメータ

ワークスペースの作成及びオープン時に以下のパラメータを指定することができます。指定方法は、”パラメータ名=値”の文字列で指定します。複数指定する場合、セミコロン(;)で各パラメータを区切ります。

(例) "name=wks;lang=UTF-8;"

パラメータ名	タイプ	値・内容
BLOCK_SIZE	ALL	ブロックサイズを指定します。指定可能なサイズは以下になります。 4096, 8192, 16384, 32768 デフォルト 4096
BEGIN_RETRY	ALL	トランザクションの開始時に他のプロセスまたはスレッドが実行している場合、再実行のためのリトライスリープ秒数を指定します。デフォルト 100
BEGIN_SLEEP	ALL	トランザクションの開始時に他のプロセスまたはスレッドが実行している場合、再実行のためのリトライカウント数を指定します。単位は msec です。デフォルト 10
CACHE_RTC	ALL	キャッシュページがすべて使用され空きがない場合、空きを待つリトライカウント数を指定します。デフォルト 100
CACHE_RTS	ALL	キャッシュページがすべて使用され空きがない場合、空きを待つリトライスリープ秒数を指定します。単位は msec です。デフォルト 10
CACHE_SIZE_DAT	ALL	データ領域のキャッシュサイズを指定します。単位はページ数です。 デフォルト 64
CACHE_SIZE_TRN	ALL	トランザクション領域のキャッシュサイズを指定します。 単位はページ数です。デフォルト 64
DBKEY	ALL	ワークスペース認証を行う場合のDBキーを指定します。 デフォルトは 指定なし
DRV_NO	ALL	ドライバ番号を指定します。複数の IO ドライバが定義されている場合、どの IO ドライバを使用するかを指定します。*eci_g_iodrv_tbl[]の配列番号でデフォルトは 0
ENCRYPT_KEY	ALL	ワークスペースを暗号化する場合、または暗号化されている際に暗号キーを指定します。デフォルト ""
FILE_MODE	FILE	ファイル作成時のファイルモードを指定します。UNIX系の OS で指定可能です。8進表記で指定します。デフォルトは 0644
FILE_SFX_DAT	FILE	データファイル作成時の拡張子指定 デフォルト "dat"
FILE_SFX_TRN	FILE	トランザクションファイル作成時の拡張子指定 デフォルト "trn"
HASH_SIZE_DAT	ALL	データ領域のキャッシュハッシュテーブルサイズを指定します。 CACHE_SIZE_DAT を拡張した場合にその値に適した数を設定します。 デフォルト 128

HASH_SIZE_TRN	ALL	トランザクション領域のキャッシュハッシュテーブルサイズを指定します。 CACHE_SIZE_TRN を拡張した場合にその値に適した数を設定します。 デフォルト 128
HDD_NO	DEVICE	ハードディスク番号を指定します。 ハードディスク番号は0からはじまり、指定がない場合のデフォルトは 0 です。
LANG	ALL	日本語コードを指定します。指定可能な文字コードは以下。 "SJIS", "EUC", "UTF-8", "UTF-16LE" "UTF-16BE" デフォルト "SJIS"
LOCK_RETRY	ALL	ワークスペースがロックでビジーである場合のリトライカウント数を指定します。 デフォルト 100
LOCK_SLEEP	ALL	ワークスペースがロックでビジーである場合のリトライスリープ秒数を指定します。単位は msec です。デフォルト 10
★ NAME	ALL	ワークスペース名を指定します。すべてのタイプで必須パラメータです。 値は大文字・小文字を識別し、31文字以下の名前を指定してください。 タイプが "FILE" の場合、ワークスペースファイルの名前になります。その場合、原則として NAME にはファイルパスを含めないでください。パスは "PATH" に指定してください。(NAME にパスが含まれている場合、暗黙的に NAME と PATH に内部で分割します。)
OVERWRITE	ALL	ワークスペース作成時のパラメータで既に存在している場合に上書きするかどうかのパラメータです。上書きする場合は true を指定します。上書きしない場合は、パラメータを指定しないか false を指定します。デフォルト "false"
PATH	FILE	ファイルパスを指定します。原則としてパス部分だけを指定し、ファイル名は NAME に指定します。 例えば "/aaa/bbb/db" の場合 "PATH=/aaa/bbb;NAME=db;" として指定します。例外として RAW デバイスファイルは、ファイル名も含め指定します。
SECTOR_SIZE	DEVICE	セクターサイズを指定します。デフォルト 512 バイト
☆ SIZE_DAT	ALL	データ領域の最大サイズを指定します。 サイズ数の後に単位 M (MB) K(KB) 文字を指定します。 例) 10MB の指定 SIZE_DAT=10M 単位を指定しない場合は、ブロック数になります。 管理形態が単一管理の場合はこのパラメータは必須です。 また分割管理の場合は、データ領域の制限サイズになります。
☆ SIZE_TRN	ALL	トランザクション領域の最大サイズを指定します。 サイズ数の後に単位 M (MB) K(KB) 文字を指定します。 例) 10MB の指定 SIZE_TRN =10M 単位を指定しない場合は、ブロック数になります。 管理形態が単一管理の場合はこのパラメータは必須です。 また分割管理の場合は、トランザクション領域の制限サイズになります。

☆ START_SECTOR	DEVICE	データ領域の開始セクタ番号を指定します。 タイプ DEVICE の必須パラメータです。
TRAN_HASH	ALL	トランザクションハッシュテーブルサイズの指定。デフォルト 128
TYPE	ALL	ストレージタイプを指定します。タイプは"FILE", "MEMORY" "DEVICE", "SQLITE"です。先頭3文字までの一致で認識します。デフォルト"FILE"
TYPE2	FILE SQLITE	ストレージタイプ固有の特殊な管理形態を指定することができます。 FILE では"raw" "raw_dev" が指定可能です。 "raw"は、新規ファイルに単一管理します。 "raw_dev"は、ワークスペースとして RAW デバイスファイルを使用することが可能です。
WARM_RESTART	ALL	起動時に未解決のトランザクションが見つかった場合にトランザクションを解決するかどうかのフラグになります。デフォルト "true"

※★表示されたパラメータはすべてのタイプで必須です。

☆は一部タイプで必須のパラメータになります。

※背景が灰色のパラメータはチューニングパラメータになります。これらのパラメータは内部的であり、基本的には調整された値であるため、変更する場合は注意が必要となります。内容を理解した上で変更していただく必要があります。

以下のパラメータはリードオンリーで設定はできませんが、ワークスペースパラメータ取得の `egi_get_wii()` `egi_get_wis()` で参照は可能なパラメータです。

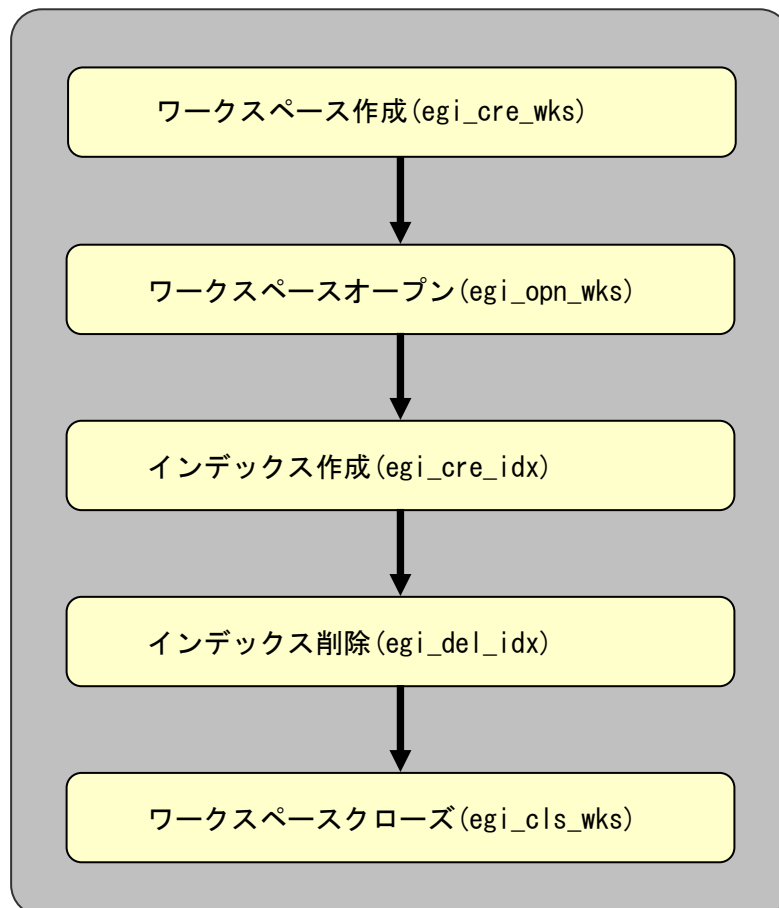
○ リードオンリーパラメータ

パラメータ名	タイプ	値・内容
ENDIAN	ALL	ワークスペースバイトオーダー デフォルト 1 0 - little endian 1 - big endian ※エンディアンの変更は、SpeeDBee/KVS ポーティングコンフィグレーションで変更
FORMAT	ALL	ワークスペースフォーマット 32bit/64bit 対応 0 - 32bit (ファイルの最大は 2GB まで) 1-64bit ※フォーマットの変更は、SpeeDBee/KVS ポーティングコンフィグレーションで変更

8.4. インデックス定義

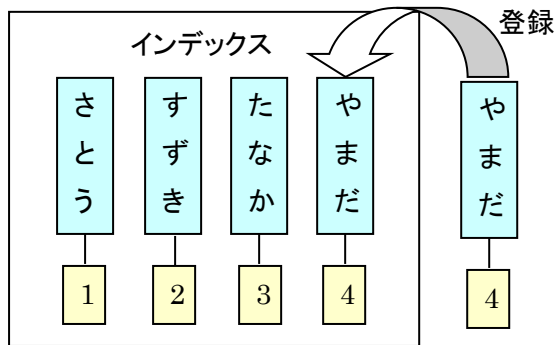
インデックスを定義します。定義内容としてインデックス種別やデータ型などがあり、検索用途に合わせてインデックスを作成してください。インデックスはワークスペース上に作成されるので、事前にワークスペースは作成する必要があります。

また、不要になったインデックスはワークスペースから削除することができます。その際、登録されているデータは全て削除されるので注意してください。

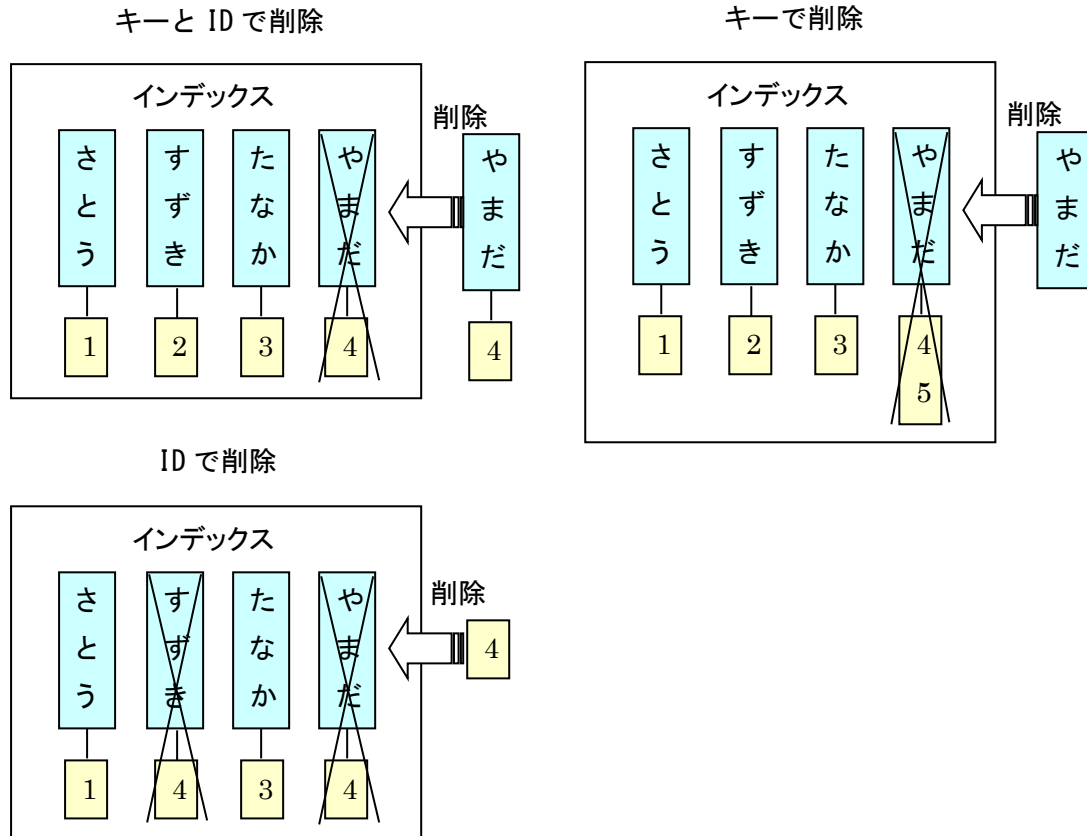


8.5. インデックス更新

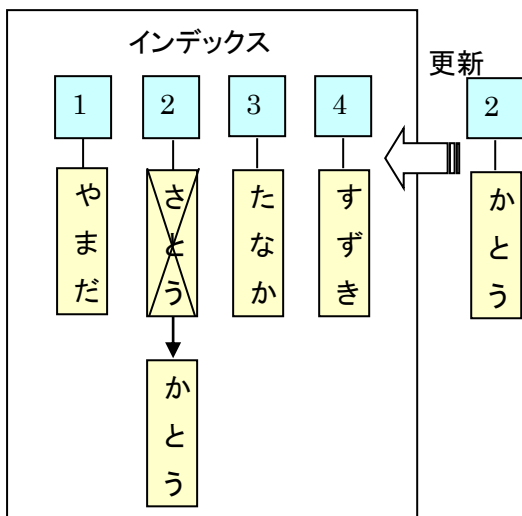
作成したインデックスに対するデータ登録、削除、更新、クリアを行います。
 データ登録は索引となるキーデータとそれに対応する ID データを登録します。



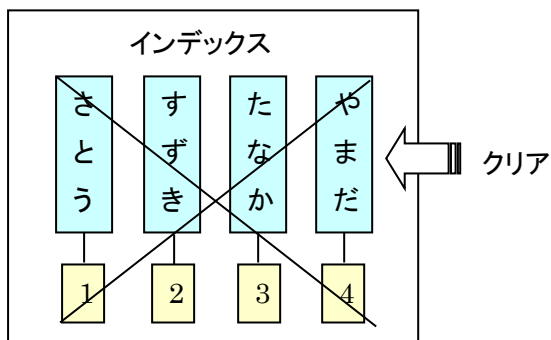
データ削除は登録したキーデータと ID データを指定することで削除します。
 キーデータのみを指定してそれに該当する ID データ全てを削除することもできます。
 また、ID データのみを指定して該当するキーデータと ID データの組み合わせで削除することができます。



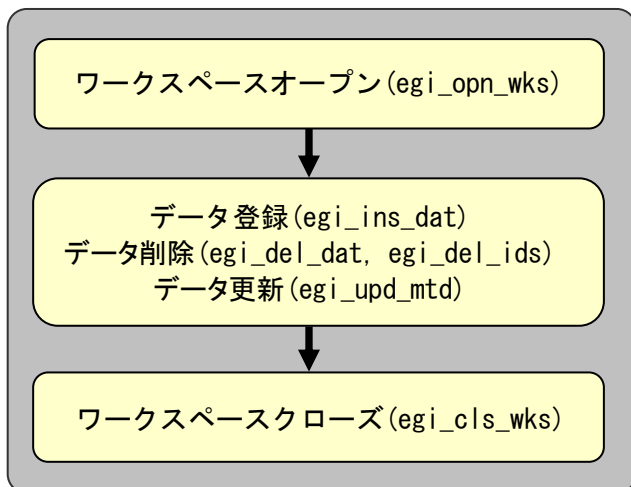
データ更新は MTREE インデックスで ID データを更新します。その他インデックスでは使用できません。



データクリアはインデックスに含まれる全てのデータを削除します。



ワークスペースオープン後、任意の更新処理を行います。



8.6. トランザクション

トランザクション機能はインデックス更新中に発生した障害に対してデータ整合性を保証します。

データ更新操作およびインデックス DDL 操作においては、内部で暗黙的にオートコミット機能が働きます。オートコミットが実行される API は以下になります。

- ・ データ更新操作

<code>egi_ins_dat</code>	データ登録
<code>egi_del_dat</code>	データ削除
<code>egi_del_ids</code>	ID データを指定した削除
<code>egi_upd_mtd</code>	M-Tree インデックス ID データの更新

- ・ インデックス DDL 操作

<code>egi_cre_idx</code>	インデックスの作成
<code>egi_del_idx</code>	インデックスの削除
<code>egi_clr_idx</code>	インデックスのクリア

また、オートコミット機能を利用せず、明示的にトランザクションの開始 (BEGIN) および終了 (COMMIT, ROLLBACK) を指示して操作することもできます。この場合、上記の操作を行う際、トランザクションを有効 (BEGIN) にすることで障害発生時にトランザクション有効範囲を元に戻す (ROLLBACK) ことができます。また、更新処理が正常に終了した場合は更新データを実データとして有効にします (COMMIT)。

※オートコミットの場合、1回の操作につき1トランザクションが実行されるため、更新処理の量が多い場合は、明示的にトランザクションを実行し、まとめて更新したほうがパフォーマンスは向上します。

明示的なトランザクションは、ワークスペースオープン後からワークスペース終了までの間であれば、使用することができます。

※但し、外部リストインデックス (インメモリのテンポラリ) はトランザクションの対象ではありません。

トランザクション利用の注意ですが、トランザクションリソースはワークスペースに対し 1 つであるため、同じワークスペースに対して、複数のプロセスまたはスレッドから同時にトランザクションをかけることはできません。(他のプロセスおよびスレッドがトランザクションを実行している際にトランザクションを実行した場合、他のプロセスおよびスレッドにより既にトランザクションが実行されているというエラーになります。)

トランザクションの分離レベルは READ COMMITTED となり、COMMIT 済みのデータに対して参照 (検索) ができます。更新中のデータに関しては他のスレッドからは参照できません。

※本バージョンではネストトランザクションおよびセーブポイント機能は非サポートです。

トランザクション中の障害復旧に関してはウォームリスタート機能をサポートしています。電断等でトランザクション中に障害があった場合、次回起動時に自動復旧することができます。

復旧内容は次のようになります。

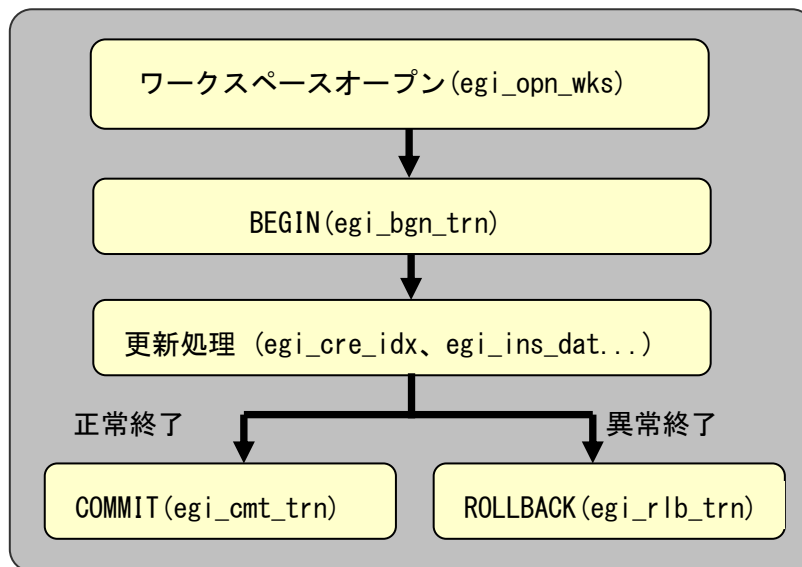
(BEGIN~COMMIT 前の電断) ROLLBACK します。

(COMMIT 中の電断) COMMIT を再開します。

(ROLLBACK 中の電断) ROLLBACK を再開します。

ウォームリスタート機能は、デフォルトで実行されます。

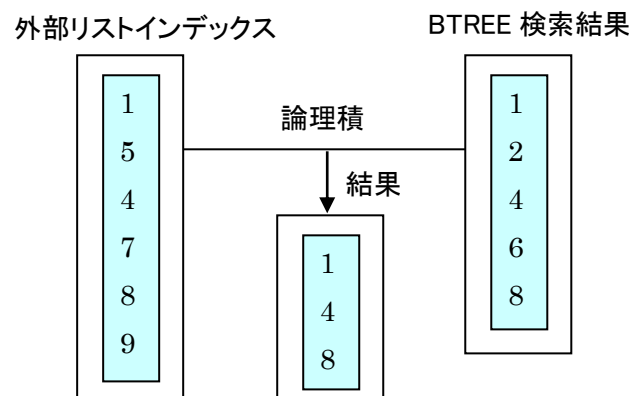
明示的なトランザクション操作に関する処理フローは次のようになります。



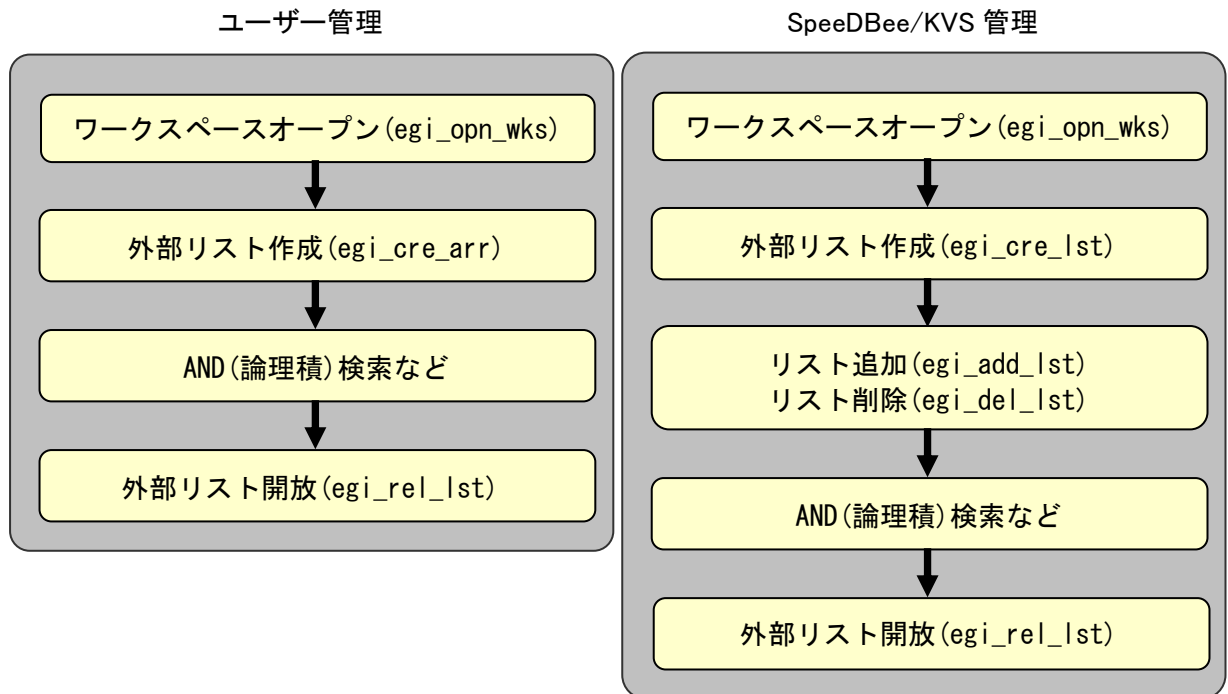
8.7. 外部リストインデックス

外部リストインデックスはユーザーが作成した ID データのリスト列を仮想インデックスとして扱います。メモリ上で管理され、記憶媒体への保存は行いません。

この外部リストインデックスは BTREE インデックスのように条件指定をした検索が出来ません。単に全件リスト取得することを目的としています。ただ、取得した全件リストとその他インデックスの論理積を求める場合に使用することができます。



外部リストインデックス作成において ID データリストをユーザー管理とする方法と SpeedBee/KVS 管理とする方法があります。ユーザー管理の場合、ID データリストのメモリ領域はユーザーメモリ領域となり、SpeedBee/KVS 管理の場合、SpeedBee/KVS 内部のメモリ領域を使用します。管理方法で使用方法が異なります。処理フローは次のようになります。



8.8. インデックス検索

各インデックス種別により可能な検索パターンの制限はありますが、次のような検索が行えます。

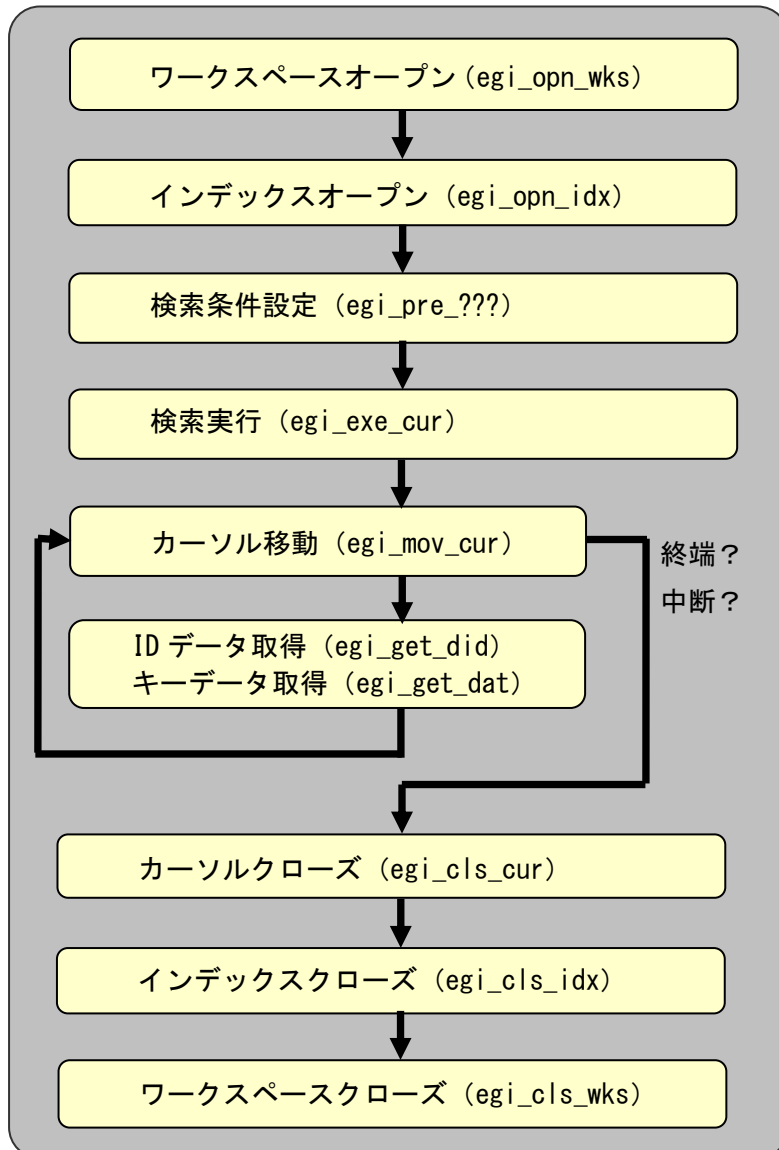
- ・ 完全一致
- ・ 前方一致
- ・ 中間一致
- ・ 空間検索
- ・ インクリメント検索

インクリメント検索を除く基本処理フローは次のようになります。

データ検索を行う場合、検索対象インデックスをオープンする必要があります。インデックスオープン後、検索条件を指定し、検索実行します。検索実行中は中断処理もできます。

検索実行後、必要に応じて ID データまたはキーデータを取ります。カーソルが検索リストの終端に到達した場合、またはユーザーによるリスト取得の中断が行われた場合、最後にカーソルをクローズします。

インデックス使用後はオープンされているインデックス及びワークスペースをクローズします。



インクリメント検索は上記基本フローで検索条件設定の前に特殊な処理を行う必要があります。インクリメント検索を行う場合、検索対象インデックスをオープンする必要があります。

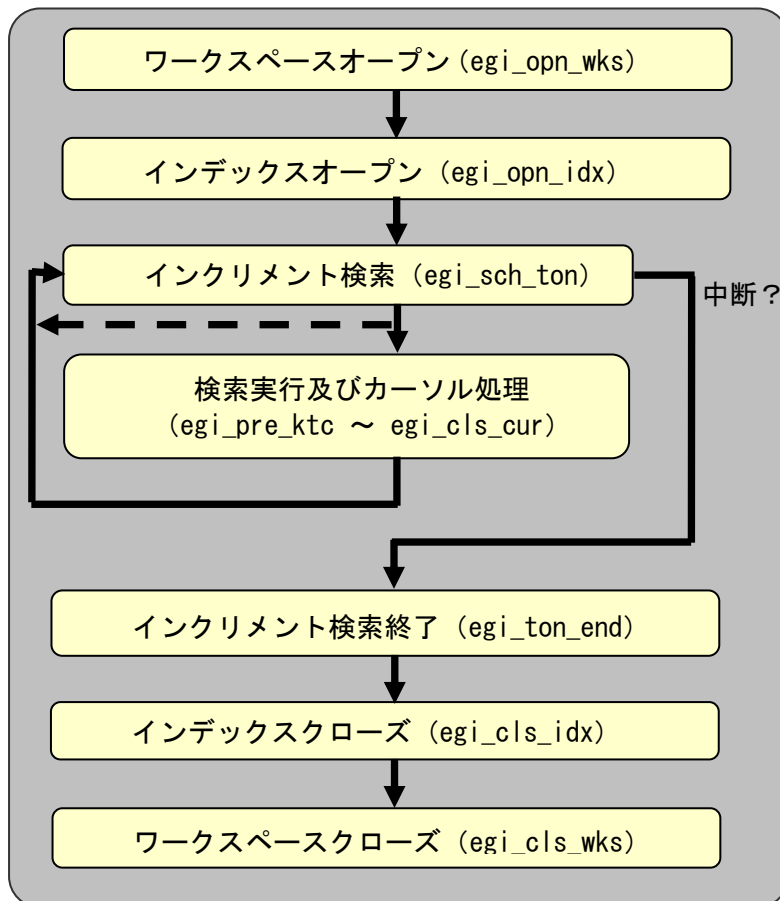
インデックスオープン後、インクリメント検索(絞込み)を行います。

結果リストを取得する場合、絞込み結果を元にカーソルオープンします。

カーソル処理は前述と同様の手順になります。

カーソル処理終了後、必要に応じて次のインクリメント検索をします。尚、リスト取得を行わない場合、カーソル処理は必要ありません。インクリメント検索を順次行ってください。

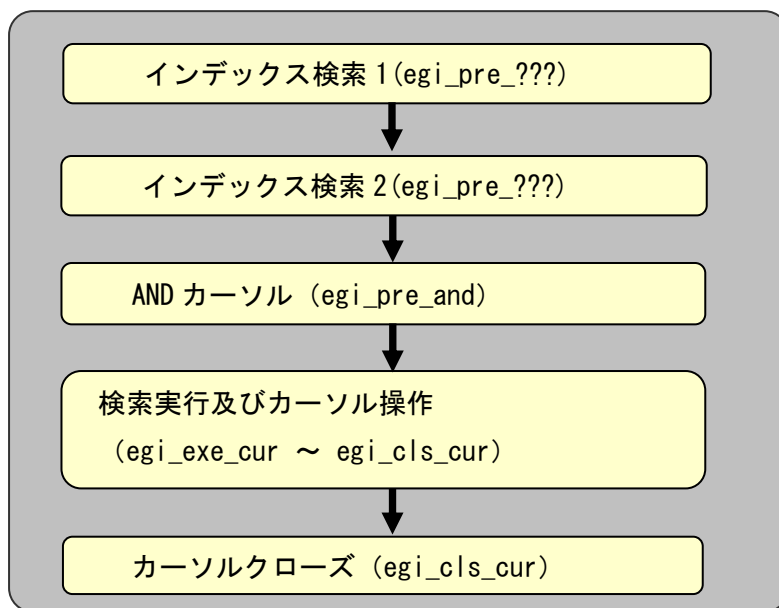
インクリメント検索終了の際は明示的に検索終了(egi_ton_end)を呼び出す必要があります。



上記のインデックス単体での処理に加え、外部リストインデックスを含む各インデックスでの検索結果を AND マージすることができます。

論理積を求める対象となるカーソルを 2 つ準備し、AND カーソルを行います。作成されたカーソルに対して更に別のカーソルを AND することも可能です。

最終的に作成された AND カーソルをクローズする際、個別で作成されたカーソルも全てクローズされます。



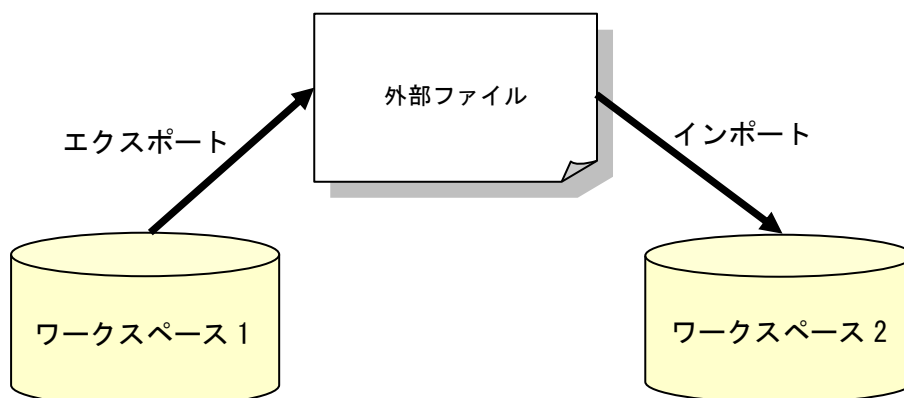
8.9. インデックス学習補助機能

文字列型インデックスに対して更新、検索を行う際の文字列変換機能及び並び替え機能をサポートします。

文字列変換機能では1つの例として英字の大文字、小文字を統一してインデックス学習させることができます。その際の文字列変換パターンは更新系と検索系で統一されます。並び替え機能はインデックス内のキーデータに対する並び替えが可能になります。並び替えをユーザー意向とすることで、検索結果取得時にユーザー指定のデータ並び順でキー及び ID を取得することができます。

8.10. メンテナンス

ワークスペースとインデックスに対するエクスポート／インポート機能をサポートします。エクスポートはワークスペースとインデックスのスキーマ情報及びインデックスデータを外部ファイルに出力します。インポートは出力された外部ファイル情報に基づいてワークスペース作成やインデックス作成などを行います。



インポート時に指定ワークスペースが存在しない場合、外部ファイル情報に基づいて新規でワークスペース及びインデックスを作成します。インポート先が既に存在するワークスペースの場合、そのワークスペースに対してインデックス作成、データ登録を行います。但し、インデックスが既に作成されている場合、データ登録のみを行うことになります。また、既存インデックスにデータが存在した場合、既存データを全てクリアしてから入力データを登録します。

既存ワークスペースに対するインポートは制限事項があります。下記の場合、エラーとなります。

- ・ワークスペース情報に設定されている文字コード(EGI_WKS_INFO.lang)が違う。
- ・インデックス情報に設定されている重複 ID 管理(EGI_IDX_INFO.otype)を除く全てのパラメータで合致しない項目がある。

インポート時に異常終了した場合、インポート前の初期状態に戻ることはないので注意してください。

本機能はバックアップ的な要素とは別にワークスペースの最適化にも有効な手段となります。更新処理を繰り返したワークスペースに対して本機能を使用することでインデックス内の断片化が修復されます。

その他注意事項として本機能を使用する場合、不揮発性の記憶媒体にあるワークスペースを対象とします。揮発性メモリのような記憶装置に存在するワークスペースに関しては使用できません。また、エクスポート／インポート時は対象ワークスペースが未運用の状態で行ってください。運用状態での動作は保証しません。

9.暗号化

9.1. 暗号化の対象

SpeedBee/KVS ではワークスペース全体が暗号化の対象となり、暗号化を行う単位はブロック単位となります。暗号化を行うモジュールは、SpeedBee/KVS の本体には持ちません。

I/O ドライバ、システムドライバと同様にインターフェースのみの定義であり、暗号化ドライバはユーザーの実装となります。

実装方法については「SpeedBee/KVS SpeedBee/RDB version5.0 ポーティングマニュアル 3.2 暗号化ポーティング」を参照してください。

データの暗号化はワークスペース作成時に暗号化するかどうかの指定を行います。

ワークスペースパラメータに“ENCRYPT_KEY”の指定があり、かつ、暗号化ドライバが実装されていれば暗号化を行います。暗号化指定は作成時のみ可能であり、暗号化されていないワークスペースを途中から暗号化することはできません。

暗号化されたワークスペースのオープン時に以下のチェックを行い、条件に満たない場合、アクセスはできず、暗号に関するユーザーエラーとなります。

- ・暗号化ドライバが実装されているかどうか。
- ・暗号化ドライバが作成時に使われた暗号化ドライバと一致するかどうか。
- ・オープンパラメータに“ENCRYPT_KEY”が指定されているかどうか
- ・ENCRYPT_KEY が間違っていないかどうか。

10. ユーザー定義データ型

SpeedBee/KVS は、キーデータタイプとして例えば「時間+ID」のような複合した値をキーとする場合、対応するデータタイプが存在しない(※1)ため、ユーザーが定義可能なデータタイプをサポートします。

ユーザー定義データタイプに対応するインデックスタイプは、MTREE, BTREE の2種類です。

※1 インデックスキーを複数指定が可能な複合インデックスという機能がありますが、NGRAM のみの対応でその他のインデックスタイプではサポートしていません。

10.1. ユーザー定義データ型の定義

ユーザー定義データ型は、ユーザーが定義するデータタイプでインデックスキーとして使用することができます。

指定可能なデータ定義は、以下の規則があります。

- 固定長データであること
- 最大サイズ 128 バイト

10.2. ユーザー定義データ型のインデックス作成

ユーザー定義データ型のインデックスの作成は、以下の2つのパラメータを指定します。

① ユーザー定義データタイプ EGI_DT_USER

ユーザー定義データタイプの定義名は EGI_DT_USER とします。

インデックス定義情報(EGI_IDX_INFO)のデータタイプパラメータ dtype に指定します。

データタイプ名	説明
EGI_DT_USER	ユーザーデータ型

② ユーザー定義データ型 ID

ユーザー定義データタイプの ID(※1)をインデックス定義情報(EGI_IDX_INFO)のパラメータ dsize に指定します。

※1 ID は、unsigned short 型で 値は 1 ~ 128 までの値を有効とします。

③ インデックス作成

MTREE インデックスの作成

```
EGI_IDX_INFO ii = {EGI_TP_MTREE, 0, 0, 0, 0, 0, EGI_DT_USER, 1, 0};
egi_cre_idx(wks, "idx1", &ii)
```

10.3. ユーザー定義データ型の関数テーブルへの登録

ユーザー定義データ型を利用する場合、システムドライバにユーザー定義テーブルおよび対象のデータ型の比較関数、コピー関数を用意する必要があります。

登録方法は「SpeedBee/KVS SpeedBee/RDB ポーティングマニュアル SpeedBee/KVS ユーザー定義データ型」を参照してください。

11. リングストレージ機能

リングストレージ機能は、限られたストレージリソースにおいて指定された最大サイズまたは最大件数を越えた場合、自動的に古いデータを削除し、登録スペースを空けることでメンテナンスなしに永続的に登録し続けられる機能です。この機能は MTREE および BTREE (UNIQUE) インデックスでリングストレージ機能を利用することができます。

11.1. リングストレージ定義の指定

リングストレージ機能を利用する場合、インデックス作成時に定義を行う必要があります。リングストレージの定義は、インデックス作成時のみ設定可能で作成した後で指定することはできません。

リングストレージ機能を付加するためのインデックス作成 API は、「8.4.2 `egi_cre_idx_ext()`」で行います。
(`egu_cre_idx()` API では付加することができません。)

`egi_cre_idx_ext()` の第4引数のインデックスオプション情報にリングストレージの定義を設定します。

インデックスオプション情報は次になります。

```
typedef struct _egi_idx_opt{
    H    type;          /* option type */
    W    par1;         /* option parameter 1 */
    W    par2;         /* option parameter 2 */
    W    par3;         /* option parameter 3 */
    W    par4;         /* option parameter 4 */
} EGI_IDX_OPT, *PEGI_IDX_OPT;
```

リングストレージ機能は、最大サイズによる制限と最大データ件数による制限の2つのタイプがあります。最大サイズの制限は、`EGI_IDX_OPT_AUTO_DEL_WKS` を指定し、最大データ件数による制限は、`EGI_IDX_OPT_AUTO_DEL_IDX` をインデックスオプション情報のパラメータ `type` に指定します。

インデックスオプション情報の `par1 ~ par4` は、指定した `type` 定義により異なります。

11.2. 最大サイズ指定 EGI_IDX_OPT_AUTO_DEL_WKS

以下は最大サイズによる制限設定のパラメータになります。

パラメータ	指定内容	必須	設定例
type	EGI_IDX_OPT_AUTO_DEL_WKS	○	type=EGI_IDX_OPT_AUTO_DEL_WKS
par1	最大サイズ(制限サイズ) 1以上を指定すること。 par3 に以下の最大サイズの単位を指定する。	○	最大サイズ 100MB par1 = 100 を指定
par2	自動削除時に削除するサイズ 1以上を指定すること。 par4 に以下の最大サイズの単位を指定する。	○	削除サイズ 1MB par2 = 1 を指定
par3	par1 に指定した最大サイズの単位を指定。MB か GB のみ有効 EGI_IDX_OPT_PAR_UNIT_MB MB EGI_IDX_OPT_PAR_UNIT_GB GB	○	par1 の単位は MB par3 = EGI_IDX_OPT_PAR_UNIT_MB を指定
par4	par2 に指定した削除サイズの単位、以下のいずれかを指定 EGI_IDX_OPT_PAR_UNIT_BC ブロック数 EGI_IDX_OPT_PAR_UNIT_KB KB EGI_IDX_OPT_PAR_UNIT_MB MB EGI_IDX_OPT_PAR_UNIT_GB GB	○	par2 の単位は MB par4 = EGI_IDX_OPT_PAR_UNIT_MB を指定

サイズ指定の自動削除実行のトリガーとなるサイズは、API 内部で決められており、par1 で指定したサイズから 10 ブロック分(ページ)を引いたサイズになります。

11.3. 最大件数指定 EGI_IDX_OPT_AUTO_DEL_IDX

最大件数による制限は、指定された件数内で動作する設定になります。

パラメータ	指定内容	必須	設定例
type	EGI_IDX_OPT_AUTO_DEL_IDX	○	type= EGI_IDX_OPT_AUTO_DEL_IDX
par1	最大件数 1以上を指定すること。	○	最大件数 10,000 件 par1 = 100000 を指定
par2	自動削除時に削除する件数 1以上を指定すること。	○	削除件数 10000 件 par2 = 10000 を指定
par3	指定不要	×	
par4	指定不要	×	

SpeeDBee/KVS Index 機能ユーザガイド

Copyright (C) 2020 SALTYSTER Inc. All rights reserved.

本書の記載内容の著作権は、株式会社ソルティスターに帰属します。

内容の全部か一部を問わず、著作権の許可なく転載、複製することを禁じます。