

# Bibliotheca2 TextSearch Version 2 プログラマーズガイド

解説・文法書

3000-7-225-40

## マニュアルの購入方法

このマニュアル，および関連するマニュアルをご購入の際は，  
巻末の「ソフトウェアマニュアルのサービス ご案内」をご参  
照ください。

## 対象製品

適用 OS : AIX 5L V5.1 , V5.2 , V5.3

P-1MD3-4711 Bibliotheca2 TextSearch Server Library Version 2 02-82\*

適用 OS : HP-UX11i(PA-RISC) , HP-UX 11i V2(PA-RISC)

P-1BD3-4711 Bibliotheca2 TextSearch Server Library Version 2 02-82\*

適用 OS : Red Hat Enterprise Linux AS 4(x86) , Red Hat Enterprise Linux ES 4(AMD64 & Intel EM64T)

P-9SD3-4711 Bibliotheca2 TextSearch Server Library Version 2 02-82\*

適用 OS : Solaris8(SPARC) , Solaris 9(SPARC)

R-18956-47 Bibliotheca2 TextSearch Server Library Version 2 02-82\*

適用 OS : Windows 2000 , Windows Server 2003 , Windows Server 2008 , Windows XP , Windows Vista , Windows 7

P-24D3-4414 Bibliotheca2 TextSearch Development Kit Version 2 02-82

P-24D3-4514 Bibliotheca2 TextSearch Runtime Library Version 2 02-82

P-24D3-4714 Bibliotheca2 TextSearch Server Library Version 2 02-82

\* 印のプログラムプロダクトについてはサポート時期をご確認ください。  
上記のプログラムプロダクトのほかにもこのマニュアルをご利用になれる場合があります。詳細は「リリースノート」でご確認ください。

## 輸出時の注意

本製品を輸出される場合には、外国為替および外国貿易法ならびに米国の輸出管理関連法規などの規制をご確認の上、必要な手続きをお取りください。

なお、ご不明な場合は、弊社担当営業にお問い合わせください。

## 商標類

AIX は、米国における米国 International Business Machines Corp. の登録商標です。

HP は、米国 Hewlett-Packard Company の会社名です。

HP-UX は、米国 Hewlett-Packard Company のオペレーティングシステムの名称です。

Linux は、Linus Torvalds の米国およびその他の国における登録商標あるいは商標です。

Microsoft は、米国およびその他の国における米国 Microsoft Corp. の登録商標です。

Microsoft Internet Explorer は、米国 Microsoft Corp. の商品名称です。

PA-RISC は、米国 Hewlett-Packard Company の商標です。

Red Hat は、米国およびその他の国で Red Hat, Inc. の登録商標若しくは商標です。

Solaris は、米国 Sun Microsystems, Inc. の米国およびその他の国における商標または登録商標です。

UNIX は、X/Open Company Limited が独占的にライセンスしている米国ならびに他の国における登録商標です。

Visual C++ は、米国およびその他の国における米国 Microsoft Corp. の登録商標です。

Windows は、米国およびその他の国における米国 Microsoft Corp. の登録商標です。

Windows Server は、米国 Microsoft Corporation の米国及びその他の国における登録商標です。

Windows Vista は、米国 Microsoft Corporation の米国及びその他の国における登録商標です。

## 発行

2001年7月(第1版)3000-7-225

2009年12月(第5版)3000-7-225-40

## 著作権

All Rights Reserved. Copyright (C) 2001, 2009, Hitachi, Ltd.

## 変更内容

変更内容 ( 3000-7-225-40 ) Bibliotheca2 TextSearch Development Kit Version 2 02-82 ,  
Bibliotheca2 TextSearch Server Version 2 02-82 , Bibliotheca2 TextSearch Runtime Library  
Version 2 02-82 , Bibliotheca2 TextSearch Server Library Version 2 02-82

追加・変更内容	変更箇所
次の関数を追加した。 • TS2F_well_formed_xmlcpl • TS2F_well_formed_xmlcpl_free	3.1 , 3.5.15 , 3.5.16
Library を新規にインストールした場合のデフォルトインストールパスを変更した。	付録 A.1 , A.2
指定できる文字コード種別に ucs4 範囲の UTF-8 を追加した。	付録 A.3 , 付録 C , 付録 D
JIS と MS の仕様差異に対応する UTF-8 文字コードを追加した。	付録 C
TS2XMLenv コマンドを実行する場合の環境変数について追加した。	付録 D
TS2XMLenv コマンドを実行する場合に Cosminexus Version 7 以降を使用するときの注意を追加した。	付録 D
関数のエラーメッセージを追加した。 • KIBS2088-E	付録 D
関数のエラーステータスを追加した。 -130045 , -130055 , -130056 , -130057 , -130058 , -130059 , -130060 , -130061 , -130062 , -130063	付録 E
TS サーバーのエラーコードを追加した。 • 2090 • 2091	付録 F
次の用語の説明を追加した。 • 簡易 XML 文書解析機能	付録 G
適用 OS に Windows Server 2008 , Windows Vista , Windows 7 を追加した。	-

単なる誤字・脱字などはお断りなく訂正しました。

変更内容 ( 3000-7-225-30 ) Bibliotheca2 TextSearch Development Kit Version 2 02-70 ,  
Bibliotheca2 TextSearch Server Version 2 02-80 , Bibliotheca2 TextSearch Runtime Library  
Version 2 02-80 , Bibliotheca2 TextSearch Server Library Version 2 02-80

追加・変更内容
メモリ常駐化機能をサポートした。
除外文字検索をサポートした。
通信ソケットの環境を設定できるようにした。
MINCREMENTAL ( テキストデータベースのインクリメンタル ( 最適化 ) ) に注意事項を追加した。
関数の注意事項を変更した。

---

追加・変更内容

---

関連情報ファイル名の説明を変更した。

---

次のメッセージを追加した。

- 2089
- 

次の用語の説明を追加した。

- 除外文字検索
  - アクション単位ソケット
  - セッション単位ソケット
  - メモリ常駐化定義ファイル
- 

変更内容 ( 3000-7-225-20 ) Bibliotheca2 TextSearch Development Kit Version 2 02-70 ,  
Bibliotheca2 TextSearch Server Version 2 02-70 , Bibliotheca2 TextSearch Runtime Library  
Version 2 02-70 , Bibliotheca2 TextSearch Server Library Version 2 02-70

---

追加・変更内容

---

XML 文書をサポートした。

---

半角アルファベット展開をサポートした。

---

文字コード種別に UTF-8 を追加した。

---

TS2XMLenv コマンドを追加した。

---

適用 OS に AIX , Solaris , Linux を追加した。

---



# はじめに

---

このマニュアルは、次に示すプログラムプロダクトの機能と使い方および Bibliotheca2 TextSearch 上で動作するユーザープログラムの作成手順について説明したものです。

P-24D3-4414 Bibliotheca2 TextSearch Development Kit Version 2

P-24D3-4514 Bibliotheca2 TextSearch Runtime Library Version 2

P-24D3-4714 Bibliotheca2 TextSearch Server Library Version 2

P-1BD3-4711 Bibliotheca2 TextSearch Server Library Version 2

P-1MD3-4711 Bibliotheca2 TextSearch Server Library Version 2

R-18956-47 Bibliotheca2 TextSearch Server Library Version 2

P-9SD3-4711 Bibliotheca2 TextSearch Server Library Version 2

以降、このマニュアルでは Bibliotheca2 TextSearch Server Version 2 を TS サーバーと呼びます。また、Bibliotheca2 TextSearch Version 2 の関連製品の総称を Bibliotheca2 TS と呼びます。

## 対象読者

このマニュアルは、Bibliotheca2 TS 上で動作するユーザープログラムを作成する方を対象にしています。

なお、このマニュアルは、次の内容を理解されていることを前提としております。

- Windows または UNIX に関する基礎的な知識
- C 言語に関する知識
- SGML (Standard Generalized Markup Language) に関する知識
- XML (eXtensible Markup Language) に関する知識

## マニュアルの構成

このマニュアルは、次に示す章と付録から構成されています。

### 第 1 章 解説

ユーザープログラムを作成する際の、Bibliotheca2 TS についての概要や機能について説明しています。

### 第 2 章 検索の概要

Bibliotheca2 TS での検索の種類について、例題を使って説明しています。

### 第 3 章 関数の文法

Bibliotheca2 TextSearch Development Kit Version 2 または Bibliotheca2 TextSearch Server Library Version 2 で提供している関数の機能と使い方について説明しています。

### 第 4 章 サーチコマンドの文法

Bibliotheca2 TextSearch Server Version 2 で提供しているサーチコマンドの機能と使い方について説明しています。

はじめに

#### 付録 A Library の環境設定

Library の環境設定について説明しています。

#### 付録 B ユーザープログラムの作成例

Bibliotheca2 TS 上で動作するユーザープログラムの作成例を挙げています。

#### 付録 C 検索および登録に使用できる文字

Bibliotheca2 TS での検索，およびデータ登録に使用できる文字について説明しています。

#### 付録 D XML ファイルを全文検索用データファイルに変換する

Bibliotheca2 TS で XML 文書を全文検索用データファイルに変換する方法について説明しています。

#### 付録 E 関数のエラーステータス

Bibliotheca2 TextSearch Development Kit Version 2 または Bibliotheca2 TextSearch Server Library Version 2 の関数で返されるエラーステータスについて説明しています。

#### 付録 F TS サーバーのエラーコード

Bibliotheca2 TextSearch Server Version 2 で出力するエラーコードについて説明しています。

#### 付録 G 用語解説

Bibliotheca2 TS で使用する用語について説明しています。

## 関連マニュアル

このマニュアルの関連マニュアルを次に示します。必要に応じてお読みください。

Bibliotheca2 TextSearch Version 2 システム管理者ガイド (3000-7-224)

Bibliotheca2 TS を使用する環境を設定する場合に参照してください。

Bibliotheca2 TextSearch Gateway Version 2 (3020-7-213)

Bibliotheca2 TextSearch Gateway Version 2 (3000-7-221)

Internet Explorer などのブラウザを利用して，Bibliotheca2 TS のデータベースを検索するためのインターフェースを構築する場合に参照してください。

Cosminexus XML Processor ユーザーズガイド (3020-3-E35)

XML 文書を TS サーバーに登録する場合に参照してください。

Cosminexus システム設計ガイド (3020-3-E52)

XML の最大メモリ使用量を指定する場合に参照してください。

Cosminexus リファレンス定義編 (3020-3-M11)

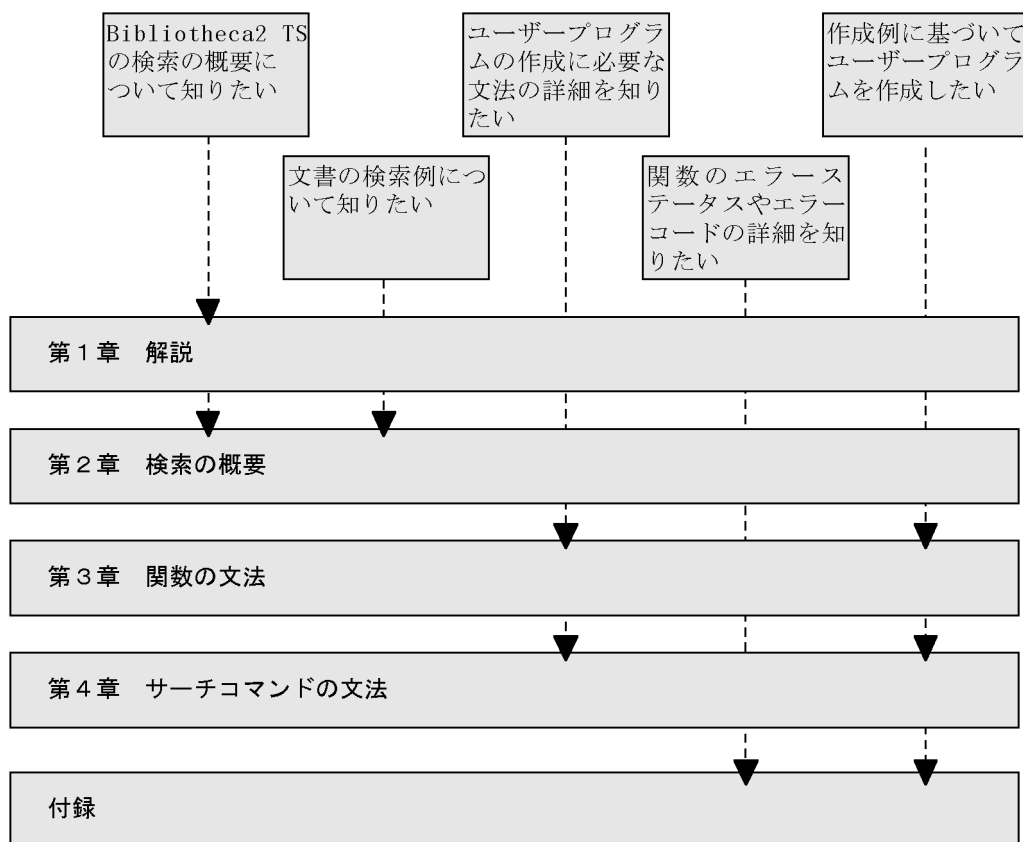
Cosminexus アプリケーションサーバ V8 リファレンス定義編 (サーバ定義)  
(3020-3-U15)

XML 文書を Cosminexus Version 7 以降を使用して登録する場合に参照してください。

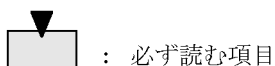


## 読書手順

このマニュアルは、利用目的に合わせて章を選択して読むことができます。利用目的別に、次の流れに従ってお読みいただくことをお勧めします。



(凡例)



## このマニュアルで使用する記号

このマニュアルで使用する記号について、次に示します。

記号	意味
	横に並べられた複数の項目に対し、項目間の区切りを示し、「または」の意味を示します。 (例) A   B   C は、「A、BまたはC」を意味します。

記号	意味
{ }	この記号で囲まれている複数の項目のうちから一つを選択することを示します。項目が横に並べられ、記号   で区切られている場合は、そのうちの一つを選択します。 (例){ A   B   C }では、A、B または C のどれかを指定することを示します。
[ ]	この記号で囲まれている項目は省略してもよいことを意味します。複数の項目が横に並べて記述されている場合には、すべてを省略するか、記号 { } と同じくどれか一つを選択します。 (例1)[ A ]は「何も指定しない」か「A を指定する」ことを示します。 (例2)[ B   C ]では「何も指定しない」か「B または C を指定する」ことを示します。
...	記述が省略されていることを示します。この記号の直前に示された項目を繰り返し複数個指定できます。 (例) A, B, B...では、「A の後ろに B を複数個指定する」ことを示します。
	1文字の半角空白を示します。
n	n 文字以上の半角空白を示します。

## このマニュアルの図中で使用する記号

このマニュアルの図中で使用する記号を次のように定義します。

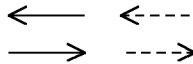
●パーソナルコンピュータ、ワークステーション



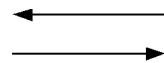
●入出力の動作



●制御の流れ



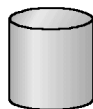
●その他の流れ



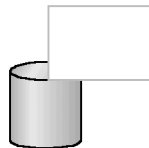
●プログラム



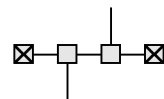
●ファイル



●ファイルの内容



●バス形のLAN



## このマニュアルで使用する構文要素

このマニュアルで使用する構文要素の種類を次に示します。

種類	定義
英字	A ~ Z a ~ z
英小文字	a ~ z
英大文字	A ~ Z
数字	0 ~ 9
英数字	A ~ Z a ~ z 0 ~ 9

種類	定義
記号	! " # \$ % & ' ' ( ) + , - . / : ; < = > @ [ ] ^ _ { } タブ 空白

## このマニュアルで使用する計算式の記号

このマニュアルで使用する計算式の記号の意味を次に示します。

記号	意味
	計算結果の値の小数点以下を切り上げを示します。 (例) $34 \div 3$ の計算結果は 12 となります。
	計算結果の値の小数点以下を切り捨てることを示します。 (例) $34 \div 3$ の計算結果は 11 となります。
max	計算結果の最も大きい値を選ぶことを示します。 (例) $\max(3 \times 6, 4 + 7)$ の計算結果は 18 となります。
min	計算結果の最も小さい値を選ぶことを示します。 (例) $\min(3 \times 6, 4 + 7)$ の計算結果は 11 となります。

## このマニュアルでの表記

このマニュアルでは、製品名称を次に示す略称で表記しています。

製品名称	略称	
AIX 5L V5.1	AIX	
AIX 5L V5.2		
AIX 5L V5.3		
Bibliotheca2 TextSearch Server Version 2	TS サーバー	
Bibliotheca2 TextSearch Version 2 の関連製品の総称	Bibliotheca2 TS	
Bibliotheca2 TextSearch Development Kit Version 2	Development Kit	Library
Bibliotheca2 TextSearch Runtime Library Version 2	Runtime Library	
Bibliotheca2 TextSearch Server Library Version 2	Server Library	
Groupmax Document Manager Version 3	Groupmax Document Manager	
Groupmax Document Manager Version 5		
Groupmax Document Manager Version 6		
HP-UX11i(PA-RISC)	HP-UX	
HP-UX11i V2(PA-RISC)		
Microsoft(R) Internet Explorer	Internet Explorer	
Microsoft(R) Word	Word	

はじめに

製品名称	略称	
Microsoft(R) Windows(R) 7 Professional Edition Windows	Windows 7	
Microsoft(R) Windows(R) 7 Enterprise Edition		
Microsoft(R) Windows(R) 7 Ultimate Edition		
Microsoft(R) Windows(R) 7 Professional x64 Edition		
Microsoft(R) Windows(R) 7 Enterprise x64 Edition		
Microsoft(R) Windows(R) 7 Ultimate x64 Edition		
Microsoft(R) Windows(R) 2000 Advanced Server Operating System	Windows 2000	
Microsoft(R) Windows(R) 2000 Professional Operating System		
Microsoft(R) Windows(R) 2000 Server Operating System		
Microsoft(R) Windows Server(R)2003 , Enterprise Edition	Windows Server 2003	
Microsoft(R) Windows Server(R)2003 , Standard Edition		
Microsoft(R) Windows Server(R)2003 R2 , Enterprise Edition		
Microsoft(R) Windows Server(R)2003 R2 , Standard Edition		
Microsoft(R) Windows Server(R)2003 , Enterprise X64 Edition		
Microsoft(R) Windows Server(R)2003 , Standard X64 Edition		
Microsoft(R) Windows Server(R)2003 R2 , Enterprise X64 Edition		
Microsoft(R) Windows Server(R)2003 R2 , Standard X64 Edition		
Microsoft(R) Windows Server(R)2008 , Enterprise Edition		Windows Server 2008
Microsoft(R) Windows Server(R)2008 , Standard Edition		
Microsoft(R) Windows Server(R)2008 , Enterprise x64 Edition		
Microsoft(R) Windows Server(R)2008 , Standard x64 Edition		
Microsoft(R) Windows Server(R)2008 R2 , Enterprise x64 Edition		

製品名称	略称
Microsoft(R) Windows Server(R)2008 R2 , Standard x64 Edition	
Microsoft(R) Windows(R) XP Professional Operating System	Windows XP
Microsoft(R) Windows(R) XP Professional x64 Edition	
Microsoft(R) Windows Vista(R) Business	Windows Vista
Microsoft(R) Windows Vista(R) Enterprise	
Microsoft(R) Windows Vista(R)Ultimate	
Microsoft(R) Windows Vista(R) Business x64 Edition	
Microsoft(R) Windows Vista(R) Enterprise x64 Edition	
Microsoft(R) Windows Vista(R) Ultimate x64 Edition	
Microsoft(R) Visual C++(R)	Visual C++
Red Hat Enterprise Linux AS 4(x86)	Linux
Red Hat Enterprise Linux ES 4(AMD64&Intel EM64T)	
Solaris 8(SPARC)	Solaris
Solaris 9(SPARC)	

### 常用漢字以外の漢字の使用について

このマニュアルでは、常用漢字を使用することを基本としていますが、次に示す用語については、常用漢字以外の漢字を使用しています。

個所（かしょ）

### KB（キロバイト）などの単位表記について

1KB（キロバイト）、1MB（メガバイト）、1GB（ギガバイト）、1TB（テラバイト）はそれぞれ1,024バイト、1,024<sup>2</sup>バイト、1,024<sup>3</sup>バイト、1,024<sup>4</sup>バイトです。



# 目次

<b>1</b>	<b>解説</b>	<b>1</b>
1.1	Bibliotheca2 TS の概要	2
1.1.1	Bibliotheca2 TS の目的	2
1.1.2	Bibliotheca2 TS を構成するプログラム	2
1.2	Bibliotheca2 TS の特長	6
1.2.1	文書の管理	6
1.2.2	文書の検索	6
1.2.3	メモリ常駐化機能	7
1.3	Bibliotheca2 TS での検索と Library	9
<b>2</b>	<b>検索の概要</b>	<b>11</b>
2.1	Bibliotheca2 TS での検索	12
2.1.1	テキストデータベースの検索方法	12
2.1.2	検索の条件の指定	13
2.2	いろいろな検索と条件式の例題	28
<b>3</b>	<b>関数の文法</b>	<b>35</b>
3.1	関数の概要	36
3.2	使用するファイル	37
3.2.1	OS が Windows の場合	37
3.2.2	OS が UNIX の場合	37
3.3	関数の発行順序	39
3.3.1	TS2F_cmd 関数でサーチコマンドを処理する	39
3.3.2	TS2F_sndcmd 関数および TS2F_rcvdat 関数でサーチコマンドを処理する	40
3.4	UAP を作成する上での注意	42
3.4.1	OS が Windows の場合	42
3.4.2	OS が UNIX の場合	43
3.5	Development Kit または Server Library で使用する関数	45
3.5.1	説明する項目	45
3.5.2	TS2F_cancelcmd (サーチコマンドの中断)	45
3.5.3	TS2F_canceldat (サーチコマンドの実行結果の破棄)	47
3.5.4	TS2F_cmd (サーチコマンドの実行)	49

3.5.5	TS2F_cmd_DATA (入力データ付きサーチコマンドの実行)	52
3.5.6	TS2F_con (通信セッションの確立)	55
3.5.7	TS2F_discon (通信セッションの切断)	56
3.5.8	TS2F_geterror (詳細エラー情報の取得)	57
3.5.9	TS2F_gettermpos (検索タームのヒット位置の取得)	59
3.5.10	TS2F_mlookcnv (サーチコマンド MLOOK の実行結果の変換)	61
3.5.11	TS2F_rcvdat (サーチコマンドの実行結果の取得および終了の確認)	63
3.5.12	TS2F_sesopt (セッションオプションの操作)	68
3.5.13	TS2F_sndcmd (サーチコマンドの実行の要求)	70
3.5.14	TS2F_sndcmd_DATA (入力データ付きサーチコマンドの実行の要求)	72
3.5.15	TS2F_well_formed_xmlcpl (整形形式 XML 文書の全文検索データ変換)	74
3.5.16	TS2F_well_formed_xmlcpl_free (整形形式 XML 文書の全文検索データ格納領域解放)	78

## 4

### サーチコマンドの文法 79

4.1	サーチコマンドの概要	80
4.2	サーチコマンドの使用上の注意	81
4.3	サーチコマンド	82
4.3.1	説明する項目	82
4.3.2	MCONCEPT (概念検索)	82
4.3.3	MDATAMAKE (文書の追加)	90
4.3.4	MDATARM (文書の削除)	94
4.3.5	MDBINF (テキストデータベース情報の出力)	95
4.3.6	MDISPLAY (検索結果の文書の出力)	102
4.3.7	MEND (テキストデータベースの使用終了宣言)	106
4.3.8	MFIND (全文検索)	107
4.3.9	MFIND (範囲指定検索)	116
4.3.10	MINCREMENTAL (テキストデータベースのインクリメンタル (最適化))	120
4.3.11	MLOOK (検索タームの同義語・異表記展開の出力)	121
4.3.12	MOPT (検索オプションの指定)	124
4.3.13	MOSTAT (検索オプションの設定値の取得)	126
4.3.14	MSORT (検索結果のソート)	128
4.3.15	MSTART (テキストデータベースの使用開始宣言)	132
4.3.16	MSWEEP (検索結果集合の削除)	133



<b>付録</b>	<b>135</b>
付録 A Library の環境設定	136
付録 A.1 OS が Windows の場合の環境設定	136
付録 A.2 OS が UNIX の場合の環境設定	136
付録 A.3 文字コード種別の環境設定	137
付録 A.4 通信ソケットの環境設定	137
付録 B ユーザープログラムの作成例	139
付録 C 検索および登録に使用できる文字	147
付録 D XML ファイルを全文検索用データファイルに変換する	152
付録 E 関数のエラーステータス	162
付録 E.1 エラーステータスの分類	162
付録 E.2 エラーステータスの形式	162
付録 E.3 エラーステータスの詳細	163
付録 F TS サーバーのエラーコード	174
付録 F.1 エラーコードの形式	174
付録 F.2 エラーコードの詳細	174
付録 G 用語解説	200

<b>索引</b>	<b>205</b>
-----------	------------

## 目次

図 1-1	OS が Windows の場合の Bibliotheca2 TS システムの構成例	4
図 1-2	OS が UNIX の場合の Bibliotheca2 TS システムの構成例	5
図 2-1	Bibliotheca2 TS での検索の概要	13
図 2-2	ランキング検索でのスコアの算出（検索ターム同士の論理演算検索の場合）に使用する例題	20
図 2-3	ランキング検索でのスコアの算出（近傍条件検索の場合）に使用する例題	21
図 2-4	ランキング検索でのスコアの算出（検索条件式同士の論理演算検索の場合）に使用する例題	22
図 2-5	概念検索の概要	25
図 2-6	例題に使用するテキストデータベース（検索対象ファイル）に格納された文書の構造	28
図 2-7	例題に使用するテキストデータベース（関連情報ファイル）に格納された文字列の構造	28
図 3-1	ファイルの格納位置	37
図 3-2	ファイルの格納位置	38
図 3-3	TS2F_cmd 関数の処理手順	39
図 3-4	TS2F_sndcmd 関数および TS2F_rcvdat 関数の処理手順	41
図 B-1	ユーザープログラムの作成例（1/8）	139
図 B-2	ユーザープログラムの作成例（2/8）	140
図 B-3	ユーザープログラムの作成例（3/8）	141
図 B-4	ユーザープログラムの作成例（4/8）	142
図 B-5	ユーザープログラムの作成例（5/8）	143
図 B-6	ユーザープログラムの作成例（6/8）	144
図 B-7	ユーザープログラムの作成例（7/8）	145
図 B-8	ユーザープログラムの作成例（8/8）	146
図 D-1	全文検索用データ変換定義ファイルの記述例（OS が Linux の場合）	155

## 表目次

表 2-1	検索タームに指定できるワイルドカードと特殊文字	14
表 2-2	同義語・異表記の展開例	16
表 2-3	ランキング検索でのスコアの算出方法（検索ターム同士の論理演算検索の場合）	20
表 2-4	ランキング検索でのスコアの算出方法（近傍条件検索の場合）	21
表 2-5	ランキング検索でのスコアの算出方法（検索条件式同士の論理演算検索の場合）	23
表 2-6	検索条件式を使用した例題	29
表 2-7	複合条件式を使用した例題	31
表 2-8	特定構造検索条件式を使用した例題	32
表 2-9	範囲指定検索条件式を使用した例題	32
表 2-10	検索結果のソートの条件式を使用した例題	33
表 2-11	概念検索の条件式を使用した例題	33
表 3-1	Development Kit または Server Library で使用する関数	36
表 3-2	使用するファイル（Windows の場合）	37
表 3-3	使用するファイル（UNIX の場合）	37
表 4-1	サーチコマンドの一覧	80
表 4-2	指定できるワイルドカード，および特殊文字	110
表 4-3	除外文字検索用の特殊文字の指定例	110
表 A-1	TSLANG の指定値	137
表 A-2	通信ソケットの設定	138
表 C-1	登録するテキストデータに使用できる文字	147
表 C-2	UTF-8 コードで使用できる文字	147
表 C-3	JISX0221 と MS-Unicode の文字別全半角対応 UTF-8 文字コード	149
表 C-4	登録文字コードと検索ターム文字コードの組み合わせによるヒット有無	149
表 C-5	¥（円記号）の取り扱い	151
表 D-1	Windows 上で TS2XMLcnv コマンドを実行する場合に設定する環境変数	156
表 E-1	エラーステータスの範囲と意味	162



# 1

## 解説

この章では、Bibliotheca2 TS の概要、特長および機能について説明します。

---

1.1 Bibliotheca2 TS の概要

---

1.2 Bibliotheca2 TS の特長

---

1.3 Bibliotheca2 TS での検索と Library

---

## 1.1 Bibliotheca2 TS の概要

---

この節では、Bibliotheca2 TS の目的、および Bibliotheca2 TS を構成するプログラムについて説明します。

### 1.1.1 Bibliotheca2 TS の目的

企業やさまざまな研究機関では、知的財産として膨大な文書を所有しています。これらの文書は整然と管理されていても、必要な情報をすばやく検索することは困難です。Bibliotheca2 TS は、データベースで管理する文書（テキストデータ）を高速に検索するためのプログラムであり、Bibliotheca2 TS を利用することで膨大な文書の中から必要な情報を迅速に探し出すことができます。

Bibliotheca2 TS で管理するデータベースは、テキストデータを管理するテキストデータベースです。例えば、新聞記事を管理しているテキストデータベースから「最新技術」に関する記事を検索する場合、キーワードとして「最新技術」を指定して検索します。すると、「最新技術」という言葉を含む記事がすべて検索できます。

また、Bibliotheca2 TS で管理するテキストデータベースでは、SGML (Standard Generalized Markup Language) や XML (eXtensible Markup Language) で、見出しや段落などの構造を明確に定義した文書を管理できます。定義した構造は検索条件として利用できます。例えば、新聞記事を管理しているテキストデータベースから、「記事の見出しに『情報技術』という単語を含む記事を探す」というような検索ができます。

また、文書の構造のほかにもさまざまな条件を指定できるので、用途に合わせたきめの細かい、柔軟な検索ができます。

### 1.1.2 Bibliotheca2 TS を構成するプログラム

Bibliotheca2 TS は次に示すプログラムで構成されます。

#### Bibliotheca2 TextSearch Server Version 2

Bibliotheca2 TS のサーバープログラムです。検索エンジンとして位置づけられ、テキストデータベースを管理しています。また、検索に利用するサーチコマンドを提供しています。

#### Bibliotheca2 TextSearch Extension Version 2

Bibliotheca2 TextSearch Server Version 2 で使用できるテキストデータベースの最大容量を変更するときに利用します。

#### **!** 注意事項

テキストデータベースの容量は、インデクスのサイズを基に計算されます。

---

Bibliotheca2 TextSearch Runtime Library Version 2 ( OS が Windows の場合だけ )

TS サーバーと通信して、テキストデータベースを検索するためのランタイムライブラリです。

Bibliotheca2 TextSearch Development Kit Version 2 ( OS が Windows の場合だけ ) および Bibliotheca2 TextSearch Server Library Version 2

検索用の UAP の作成に利用する関数と、UAP の実行に必要なライブラリを提供するプログラムです。UAP を作成、および実行するクライアントマシンにインストールします。Development Kit および Server Library には、Runtime Library のモジュールが含まれます。

なお、Development Kit と Server Library ではライセンスの形態が異なるため、OS が Windows の場合は、使用する環境に応じてどちらかをインストールしてください。

Text Search Dictionary

検索に利用する同義語辞書を提供します。一般用語や企業名、地域名などが収録されていますので、ご活用ください。

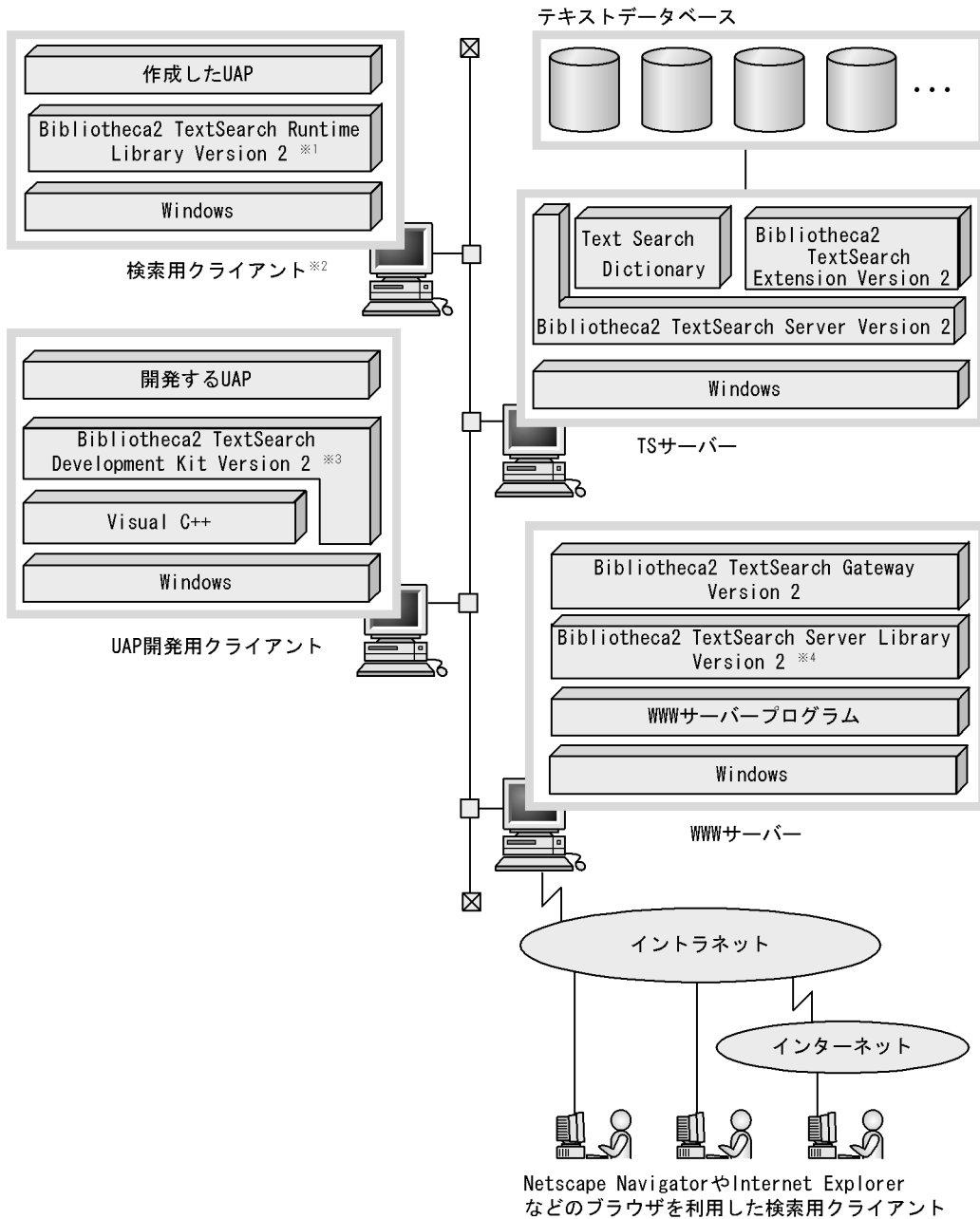
Bibliotheca2 TextSearch Gateway Version 2

インターネット、イントラネットを利用した文書検索用のインターフェースを提供します。Bibliotheca2 TextSearch Gateway Version 2 を利用すれば、UAP を作成しなくても文書を検索できます。なお、Bibliotheca2 TextSearch Gateway Version 2 を利用した検索には、Internet Explorer などのブラウザが必要です。

OS ごとの Bibliotheca2 TS システムの構成例を、図 1-1 および図 1-2 に示します。

1. 解説

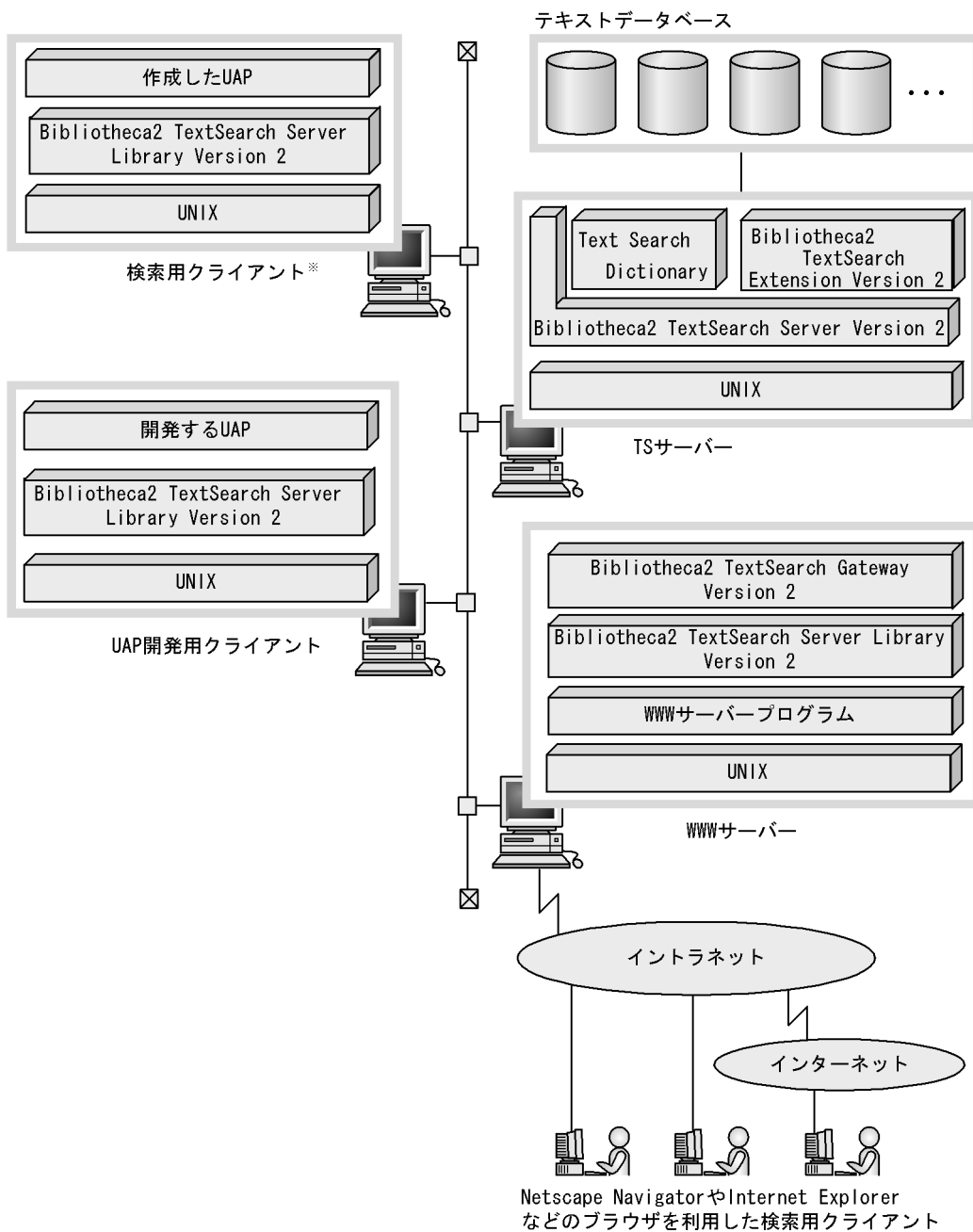
図 1-1 OS が Windows の場合の Bibliotheca2 TS システムの構成例



- 注※1 Runtime Libraryの代わりにServer LibraryまたはDevelopment Kitも使用できます。
- 注※2 UNIX上で稼働するUAPからも、文書を検索できます。
- 注※3 Development Kitの代わりにServer Libraryも使用できます。
- 注※4 Server Libraryの代わりにRuntime LibraryまたはDevelopment Kitも使用できます。



図 1-2 OS が UNIX の場合の Bibliotheca2 TS システムの構成例



注※ Windows上で稼働するUAPからも、文書を検索できます。

## 1.2 Bibliotheca2 TS の特長

---

この節では、Bibliotheca2 TS の特長について説明します。

### 1.2.1 文書の管理

Bibliotheca2 TS の文書管理には、次の二つの特長があります。

#### (1) 大量の文書の管理

Bibliotheca2 TS で管理するデータベースは、テキストデータを管理するテキストデータベースです。テキストデータを管理するので、一つのデータベースに多くのデータを格納できます。テキストデータベースは 1 サーバー当たり最大 16 個定義できます。

#### (2) テキストデータ以外の関連情報の管理

Bibliotheca2 TS は、テキストデータを文書として管理しています。しかし、通常扱うデータはテキストだけで作成されているとは限りません。見出し情報や図、表などのデータもテキストデータと同様に管理しておく必要があります。

Bibliotheca2 TS は、検索に利用するテキストデータとテキストデータ以外のデータを関連づけて管理できます。構造を持たない文書の管理には、文書情報や見出し情報などを関連情報として定義しておくとう便利です。

### 1.2.2 文書の検索

Bibliotheca2 TS での文書検索には、次の特長があります。

#### (1) ユーザー指定のキーワードによる全文検索

Bibliotheca2 TS は、文書中の単語をキーワード（検索ターム）にして文書を検索します。この検索方法を全文検索といいます。

例えば、「文書中に『最新技術』という単語が使われている文書を探す」などの検索ができます。

#### (2) 検索範囲を条件にした検索

Bibliotheca2 TS では、文書の作成日時や文書の整理番号など、前後関係のあるデータを検索キーとして、指定したデータの範囲にある文書を検索できます。この検索方法を範囲指定検索といいます。

例えば、「『2001年03月01日から2001年05月01日まで』に作成された文書を探す」などの検索ができます。

### (3) 任意の文字列や文章を手がかりとした検索

Bibliotheca2 TS では、ユーザーが任意に指定した文章や文字列を手がかりにして、その条件と似た概念を持つ文書を検索できます。この検索方法を概念検索といいます。

例えば、「『近年、環境保護に力を入れる自治体が増えている』という概念を持つ文書を探す」などの検索ができます。

### (4) SGML や XML で書かれた構造を持つ文書の管理と検索

Bibliotheca2 TS は、SGML や XML のように文書の構造を定義する言語で書かれたファイルを文書として管理できます。検索条件には、文書に定義した構造を利用できます。

例えば、「『見出し』として定義された構造内の文章中に『情報技術』という単語が存在する文書を探す」などの検索ができます。

### (5) 同義語や異表記を利用した検索

同義語辞書を作成しておくことで、検索条件として指定するキーワードと同義語辞書に定義されている同義語を基に文書を検索できます。

例えば「コンピューター」をキーワードにして検索する場合、「PC」や「パソコン」などの単語を含む文書も検索できます。同義語辞書の定義内容によっては、シソーラスを利用するような検索もできます。例えば、「インターネット」の同義語として「イントラネット」を定義しておくような利用方法も考えられます。なお、同義語を収録した Text Search Dictionary が用意されています。同義語辞書として、一般用語や企業名、地域名などが収録されていますので、ご活用ください。

また、キーワードの異表記を含む文書を検索することもできます。異表記はシステムで定義されている情報が基になります。

### (6) 「重み付け」を利用した検索

検索条件に複数のキーワードを指定する場合、それぞれの条件に対して重み（重要度）を指定できます。

例えば、「政治」と「経済」という二つのキーワードが存在する文書を検索する場合に、「経済」というキーワードより、「政治」というキーワードの重要度を上げて検索できます。この場合、「政治」の出現回数によって得点（スコア）を付けた検索結果が得られます。また、このスコアを利用して検索した文書をソート（ランキング）することもできます。

## 1.2.3 メモリ常駐化機能

メモリ常駐化機能とは、メモリを使用して登録や検索の性能を向上させる機能です。

メモリ常駐化機能を使用した場合、ディスク上に格納されているデータを入出力するためのバッファをメモリ上に確保し、使用頻度の高いデータをメモリ上に常駐化させるこ

## 1. 解説

とができます。登録や検索に使用するデータをメモリ上でやりとりすることで、文書の登録や検索性能を高速化できます。

## 1.3 Bibliotheca2 TS での検索と Library

---

ここでは Library の概要，および Bibliotheca2 TS での Library の位置づけについて説明します。

### (1) Library の概要

Library は，Bibliotheca2 TS で検索機能を使用するためのライブラリ関数を提供します。

テキストデータベースを検索するためのユーザープログラム (UAP) を，C 言語で作成する場合に利用します。また，作成した UAP の実行に必要なライブラリを提供します。UAP を作成および実行するクライアントマシンに組み込んで使用します。

### (2) 作成する UAP の概要

Library は，テキストデータベースを検索するためのコマンドを TS サーバーに発行して，実行結果を取り込む関数を提供します。テキストデータベースを検索するコマンドをサーチコマンドといいます。サーチコマンドは，UAP に記述された関数の中に指定し，テキストデータベースを検索するための条件の指定に使用します。



# 2

## 検索の概要

この章では、Bibliotheca2 TS での検索の概要について、例題とあわせて説明します。

---

2.1 Bibliotheca2 TS での検索

---

2.2 いろいろな検索と条件式の例題

---

## 2.1 Bibliotheca2 TS での検索

---

この節では、Bibliotheca2 TS での検索方法とその概要について説明します。

説明文に「参照例：」と記載している個所は、「2.2 いろいろな検索と条件式の例題」の表 2-6 ~ 表 2-11 の項番を示しています。

### 2.1.1 テキストデータベースの検索方法

Bibliotheca2 TS では、単語や文章の概念をキーワードにしてテキストデータベースを検索します。例えば、キーワードを「最新技術」として、「『最新技術』という単語を含む文書を探す」という検索ができます。

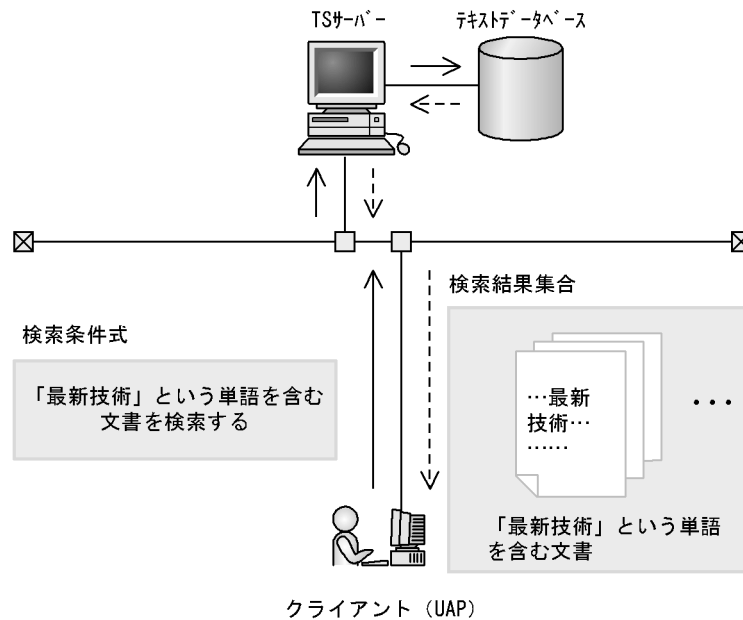
検索のキーワードになる単語を、検索タームといいます。検索タームにさらに条件を付けることで、さまざまな条件での検索ができます。また、検索の条件に文字列や文章を指定して検索することもできます。これらの検索の条件は、サーチコマンド MFIND または MCONCEPT の検索条件式に指定します。

検索条件に該当（ヒット）した文書は、検索結果集合としてグループ化されます。検索結果集合には ID（検索結果集合 ID）が付けられて管理されます。

Bibliotheca2 TS での検索の概要を次の図に示します。



図 2-1 Bibliotheca2 TS での検索の概要



(凡例)

—→ : 検索を要求する制御の流れ

---&gt; : 検索結果を返す制御の流れ

## 2.1.2 検索の条件の指定

Development Kit または Server Library を使用して作成した UAP から検索する場合、検索の条件は UAP から指定します。UAP から指定された検索条件は、UAP 内でサーチコマンド MFIND または MCONCEPT のパラメーターの条件式として受け取られ、TS サーバーに発行されます。

### (1) 検索タームを指定した検索

#### (a) 検索タームを 1 語指定した検索

『最新技術』という単語を含む文書」を検索する場合は、『最新技術』が検索タームになります。

参照例：

検索タームを 1 語指定した例：表 2-6 の項番 1

検索タームには、ワイルドカードや特殊文字も指定できます。検索タームに指定できるワイルドカードと特殊文字を次の表に示します。

## 2. 検索の概要

表 2-1 検索タームに指定できるワイルドカードと特殊文字

種類	意味	指定例	検索結果の例
*	0文字以上の任意の文字列に相当する。	文書 * 検索	文書と記事を検索 文書の検索
?	任意の1文字に相当する。 ?を続けて記述すると、続けた分だけの文字列に相当する。	概要 ? 検索	概要と検索 概要の検索
		?? システム	管理システム 昔のシステム
	指定した文字列を構造の先頭、または末尾に含むことに相当する。  文字列を囲んで指定した場合は、指定した文字列だけを構造内に含むことに相当する。	大阪府	大阪府大阪市 大阪府で開催
		株式会社	株式会社
¥	ワイルドカード、または特殊文字の意味を消して、通常の文字と同様に検索する場合に使用する(エスケープ文字)	¥* 高速 ¥? システム	* 高速 ? システム
^	直後にある1文字を検索対象から除外する。 ただし、特殊文字を¥(エスケープ文字)とともに指定した2文字は、1文字とみなして除外する	ローマ ^ 字	ローマの ローマ人
		^¥¥100	\$100 約 100
[ ]	除外する1文字を、(半角コンマ)で区切って複数指定する場合に使用する。	大阪 ^ [ 市 , 府 ]	大阪の 大阪弁
,	[ ]の中の文字を区切る場合に使用する。	文書 ^ [ の , を ] 管理	文書が管理 文書で管理

参照例：

ワイルドカードを指定した例：表 2-6 の項番 3 ~ 5

除外文字を指定した例：表 2-6 の項番 6

### (b) 検索タームを複数個指定した検索

検索条件には、検索タームを複数個指定できます。例えば、「『最先端技術』および『パソコン』の両方の単語が含まれる文書」を検索できます。この例では、検索ターム同士の論理積 (AND 条件) を検索の条件にしています。また、論理和 (OR 条件) を検索条件にして、「『最先端技術』または『パソコン』のうち、どちらかを含む文書」を検索できます。

検索タームを複数個指定する場合は、重み (重要度) も付けられます。例えば、上記の

例で「『最先端技術』に重みを付けて検索したい」という検索ができます。重みについては、「(8) 重み (重要度) を付けた検索」を参照してください。

参照例：

検索タームの論理積を指定した例：表 2-6 の項番 7

検索タームの論理和を指定した例：表 2-6 の項番 10

### (c) 検索ターム間の文字数を条件にした検索

二つの検索ターム間の文字数 (距離) を検索条件にできます。例えば、「『最新』と『技術』の間の文字数が 20 文字ちょうどの文字列を含む文書」を検索できます。このような検索を近傍条件検索といいます。

参照例：

近傍条件検索の例：表 2-6 の項番 11 ~ 14

## (2) 範囲を指定した検索

例えば、2001 年 1 月以降に作成した文書を検索したいなど、ある範囲内の文書を検索したい場合があります。範囲指定検索は、このような場合に利用します。

範囲指定検索では、日時や数字などの前後関係のあるテキストデータを検索タームに指定します。このとき、『A から B』という指定だけでなく、『A 以降』または『B 以前』という条件も指定できます。

参照例：表 2-9 の項番 1 ~ 3

## (3) 同義語・異表記展開検索

例えば「s k i」を検索タームにする場合、「ski」、「SKI」、「S k i」などで表記されている文書も検索したい場合があります。同義語・異表記展開検索はこのような場合に使用します。

同義語・異表記展開検索では、同義語辞書での定義に基づいて、検索タームの同義語が自動的に展開されます。さらに、Bibliotheca2 TS でのルールに基づいて検索タームの異表記が自動的に展開されます。展開された同義語・異表記は、検索タームと併せて検索条件にできます。

展開方法には次の 4 種類があり、これらを展開モードといいます。展開モードは組み合わせで検索条件にできます。

同義語展開

かたかな異表記展開

アルファベット異表記展開

半角アルファベット異表記展開

全角・半角異表記展開

## 2. 検索の概要

同義語・異表記の展開例を、次の表に示します。

表 2-2 同義語・異表記の展開例

	検索ター ム	展開モード 展開結果の例
同義語展開	P C	P C , コンピューター , パソコン
かたかな異表記展開	バイオリ ン	ヴァイオリン , ヴァイオリン , ヴァイオリ ン , ヴァイオリン , ヴァイオリン , ヴァ イオリン , ヴァイオリン , ヴァイオリン , ヴァ イオリン , バイオリン , バイオリン , バイオリン , バイオリン
アルファベット異表記展開	D O C	D o c , D O C , d o c
半角アルファベット異表記展開	DOC	Doc , DOC , doc
全角・半角異表記展開	T e x t	T e x t , Text
	アメリカ	アメリカ , アメリカ

### 注

「P C」の同義語が「コンピューター」、「パソコン」であると、あらかじめ同義語辞書に定義する必要があります。

なお、同義語を収録した Text Search Dictionary が用意されています。同義語辞書として、一般用語や企業名、地域名などが収録されていますので、ご活用ください。

参照例：

同義語検索の例：表 2-6 の項番 15

## (4) 検索対象の文書

検索の対象にする文書の集合をベースといいます。ベースとして、テキストデータベースに格納されているすべての文書または検索結果集合を選択できます。

ベースは、ハイアラキサーチとユニバースサーチの2種類の方法で設定します。ハイアラキサーチは、1回目の検索結果を自動的に2回目のベースとし、さらに2回目の検索結果を3回目のベースとする、という繰り返しで検索を続け、結果を絞り込む方法です。これに対して、自動的に絞り込みをしないで、ベースを固定して検索する方法をユニバースサーチといいます。例えば、テキストデータベースに格納されているすべての文書を常にベースにして検索する場合は、ユニバースサーチを選択します。

なお、ベースの設定は、サーチコマンド MOPT または MFIND で指定します。

## (5) 論理演算検索

検索条件式同士や検索結果同士の論理演算を条件にする検索を、論理演算検索といいます。例えば、「『条件 A で検索した結果』および『条件 B で検索した結果』で、両方の検索結果に含まれる文書」を検索できます。この例では、検索結果同士の論理積 (AND 条件) を検索条件にしています。さらに、論理和 (OR 条件) を検索の条件にして、「『条

件 A で検索した結果』または『条件 B で検索した結果』のうち、どちらかの検索結果に含まれる文書」を検索できます。これらの論理演算検索の条件は、サーチコマンド MFIND のオプションの複合条件式に指定します。

なお、複合条件式に検索条件式を複数指定する場合は、重み（重要度）を付けられます。重みについては、「(8) 重み（重要度）を付けた検索」を参照してください。

参照例：

検索条件式同士の論理積を指定した例：表 2-7 の項番 1

検索条件式同士の論理和を指定した例：表 2-7 の項番 4

検索結果同士の論理積を指定した例：表 2-7 の項番 5

## (6) 構造名を指定した検索

SGML や XML 文書などの構造化された文書を格納するテキストデータベースを、構造文書用テキストデータベースといいます。例えば、「文書・章・節」という構造で定義された文書を格納する構造文書用テキストデータベースがあるとします。このテキストデータベースは、「節の中に『SGML』という単語を含む文書」のように、構造を条件にして検索できます。

参照例：

構造名を指定した例：表 2-6 の項番 2, 4

### (a) 複数の構造を指定した検索

複数の構造の論理演算を検索の条件にできます。例えば、「章題および本文の中に『SGML』という単語を含む文書」を検索できます。

参照例：

複数の構造を指定した例：表 2-7 の項番 2

### (b) 繰り返し構造に対する検索

章、節などのように繰り返し現れる構造に対して条件を指定できます。例えば、章が 1 章、2 章...と繰り返している場合、章ごとに検索され、そのうちのどれかの章で条件を満たした文書がヒットします。

例えば、「章の中に『全文検索』を含む文書」を検索条件にします。この場合、「1 章の中に『全文検索』を含む文書」や「3 章の中に『全文検索』を含む文書」などがヒットします。「章の中に『全文検索』および『テキストサーチ』を含む文書」を検索条件にした場合、「全文検索」および「テキストサーチ」が同じ章の中にある文書がヒットします。

参照例：

繰り返し構造を指定した例：表 2-6 の項番 16

### (c) 特定の構造に対する検索

特定の構造を検索条件にして指定できます。この場合は、サーチコマンド MFIND のオ

## 2. 検索の概要

プシヨンの特定構造検索条件式に記述します。

次に示す構造を持つ文書を例に，特定構造検索条件式を説明します。

文書 X			
章番号	章 - 1	章番号	章 - 2
章見出し	. . A . . .	章見出し	. B . . . . .
本 文	. . . . .	本 文	. B . . . . .
. . . . . A . . . . .		. . . . . . . . . . .	
...			
文書 Y			
章番号	章 - 1	章番号	章 - 2
章見出し	. A . . . . .	章見出し	. . . . . B
本 文	. . . . .	本 文	. . . . .
. . B . . . . .		. . . . . A . . . . .	
...			

例えば、「章見出しに『A』を含み，章の本文に『B』を含む文書を検索する。ただし，『A』および『B』は同じ章の中に存在すること」を検索条件にしたとします。「同じ章の中に存在する」という条件は，章という特定の構造を条件にしています。この場合，文書 Y の章 - 1 が条件に該当するため，文書 Y がヒットします。

参照例：

特定の構造を指定した例：表 2-7 の項番 6

また，構造を特定しないで「章見出しに『A』を含み，章の本文に『B』を含む文書」と指定するとします。この場合，文書 Y と章 - 1 の章見出しに「A」を含み，章 - 2 の本文に「B」を含む文書 X もヒットします。

参照例：

複数の構造を指定した例：表 2-8 の項番 1

### (d) タグ属性に対する検索

SGML 文書，および XML 文書の構造を表したタグ属性を条件にして検索できます。例えば「属性 "security" の値が "secret" である文書」などを検索できます。

参照例：

タグ属性を指定した例：表 2-6 の項番 17

## (7) ランキング検索

### (a) ランキング検索とは

『インターネット』および『検索システム』が含まれる文書を検索したい。ただし、これらの単語が出現する回数の多い文書から取り出したい」という場合があります。このような場合は、ランキング検索を使用します。

ヒットした文書にスコア（得点）を付けてランクを付けることをランキングといいます。上記の例では、「インターネット」および「検索システム」の出現数が多い文書に高いスコアが付けられます。付けられたスコアの高い順番に検索結果をソートして、結果を取り出すこともできます。

参照例：

ランキング検索を指定した例：表 2-6 の項番 8

### (b) ランキング検索でのスコアの算出

スコアは、ヒットした文書に対して 1 ~ 100 の範囲で与えられます。検索タームを指定した検索では、指定した検索タームの出現数が多い文書ほど高いスコアが与えられます。

例えば、次の文書で検索タームを A としてランキング検索すると、検索ターム A を多く含む文書 X の方に、文書 Y よりも高いスコアが付与されます。

文書 X	文書 Y
A . . . A . .	. . . . .
. . A . . A .	. . . . . A . .
. A . . A . .	. . A . . . .
A . A . . A .	. . . . .

(凡例)

A : 検索ターム「A」

複数の検索タームの論理演算結果を条件にした場合のスコアの算出

図 2-2 に示す文書を例にして、検索ターム同士の論理演算を条件にした場合のスコアの算出方法を、次の表に示します。

## 2. 検索の概要

図 2-2 ランキング検索でのスコアの算出（検索ターム同士の論理演算検索の場合）に使用する例題

文書 X	文書 Y
. . A . . . .	. A . . . . .
. . . . . A . .	. . . . . A . .
. . A . . A .	. . A . . B .
. . . . B . . .	. . . . . B . .

(凡例)  
A : 検索ターム「A」  
B : 検索ターム「B」

表 2-3 ランキング検索でのスコアの算出方法（検索ターム同士の論理演算検索の場合）

検索条件		スコアの算出方法	説明	スコアリングの例
BOOLEAN 方式	論理積 (fand 条件)	演算対象（検索ターム）の文書内での各出現数のうち、最小値を基にスコアが付けられる。	「A および B が含まれる文書」を検索した結果、文書内での出現数が A の方が多かった場合、B の出現数を基にスコアが付けられる。	図 2-2 で「A および B が含まれる文書」を検索すると、文書 X のスコアは 50、文書 Y のスコアは 100 となる。
	論理和 (for 条件)	演算対象（検索ターム）の文書内での各出現数のうち、最大値を基にスコアが付けられる。	「A または B が含まれる文書」を検索した結果、文書内での出現数として A の方が多かった場合、A の出現数を基にスコアが付けられる。	図 2-2 で「A または B が含まれる文書」を検索すると、文書 X のスコアは 100、文書 Y のスコアは 75 となる。
スコア総和方式	論理積および論理和 (fand 条件および for 条件)	演算対象（検索ターム）の文書内でのそれぞれの出現数の合計値を基にスコアが付けられる。	「A および B が含まれる文書」、または「A または B が含まれる文書」の検索でヒットした文書内の A と B の出現数の合計を基にスコアが付けられる。	図 2-2 で「A および B が含まれる文書」または「A または B が含まれる文書」を検索すると、文書 X のスコアは 100、文書 Y のスコアは 100 となる。

(凡例)



A : 検索ターム「A」

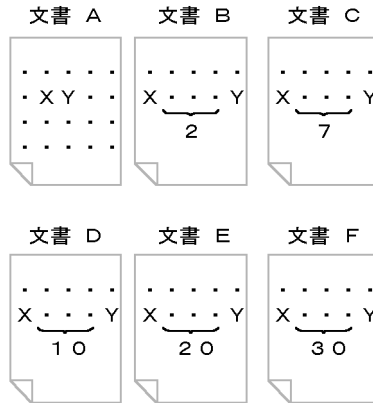
B : 検索ターム「B」

注 ここでの算出方法は、検索タームに重みが付けられていないことが前提です。検索タームに重みが付けられている場合は、算出結果が異なります。

近傍条件検索した場合のスコアの算出

次の図に示す文書を例にして、近傍条件検索した場合のスコアの算出方法を、表 2-4 に示します。

図 2-3 ランキング検索でのスコアの算出（近傍条件検索の場合）に使用する例題



(凡例)

X : 検索ターム「X」

Y : 検索ターム「Y」

{ : 検索ターム「X」と「Y」の間の文字数

表 2-4 ランキング検索でのスコアの算出方法（近傍条件検索の場合）

検索条件	スコアの算出方法	説明	スコアリングの例
二つの検索ターム間の距離（文字数）が、指定した値と一致することを条件にする。 (例) <pre>FindFile: c=10('X','Y')</pre>	指定した距離とヒットした距離が等しい場合は、100 が与えられる。	「XY 間の距離が 10 文字ちょうどの文書」を検索した結果、文書内での X と Y の距離が 10 であれば 100 が与えられる。	図 2-3 で「XY 間の距離が 10 文字ちょうどの文書」を検索すると、文書 D だけがヒットする。スコアは 100 となる。

## 2. 検索の概要

検索条件	スコアの算出方法	説明	スコアリングの例
二つの検索タームの間の距離（文字数）が指定した値以下であることを条件にする。 (例) <pre>FindFile: c&lt;=10('X','Y')</pre>	ヒットした距離が0に近いほど100に近づく。	「XY間の距離が10文字以下の文書」を検索した結果、文書内のXYの距離が0ならば100が与えられる。10以下の値は、10に近いほど低いスコアが与えられる。	図2-3で「XY間の距離が10文字以下の文書」を検索すると、文書A、B、CおよびDがヒットする。スコアの高い順に、文書A、B、C、Dとなる。
二つの検索ターム間の距離（文字数）が指定した値以上であることを条件にする。 (例) <pre>FindFile: c&gt;=10('X','Y')</pre>	指定した距離とヒットした距離が等しい場合は、100が与えられる。遠ざかるほど低くなる。	「XY間の距離が10文字以上の文書」を検索した結果、文書内のXYの距離が10ならば100が与えられる。10を超える値は、10から遠いほど低いスコアが与えられる。	図2-3で「XY間の距離が10文字以上の文書」を検索すると、文書D、EおよびFがヒットする。スコアの高い順に、文書D、E、Fとなる。

(凡例)

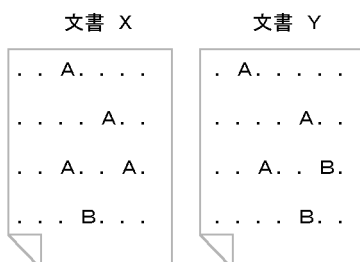
X：検索ターム「X」

Y：検索ターム「Y」

検索条件式同士の論理演算結果を条件にした場合のスコアの算出

次の図に示す文書を例にして、検索条件式同士の論理演算検索を条件にした場合のスコアの算出方法を、表2-5に示します。

図2-4 ランキング検索でのスコアの算出（検索条件式同士の論理演算検索の場合）に使用する例題



(凡例)

A：検索条件A

B：検索条件B

表 2-5 ランキング検索でのスコアの算出方法（検索条件式同士の論理演算検索の場合）

検索条件		スコアの算出方法	説明	スコアリングの例
BOOLEAN 方式	論理積 (AND 条件)	演算対象 (検索条件) の文書内で、 各出現数の うち、最小 値を基にス コアが付け られる。	「A および B が含まれる文 書」を検索した 結果、文書内 での出現数が A の方が多かった 場合、B の出 現数を基にス コアが付けら れる。	図 2-4 で「A およ び B が含まれる文 書」を検索すると、 文書 X のスコアは 50、文書 Y の スコアは 100 となる。
	論理和 (OR 条 件)	演算対象 (検索条件) の文書内 での各出現 数のうち、 最小値を基 にスコアが 付けられる。	「A または B が含まれる文 書」を検索した 結果、文書内 での出現数が A の方が多かった 場合、A の出 現数を基にス コアが付けら れる。	図 2-4 で「A また は B が含まれる 文書」を検索 すると、文書 X のスコアは 100、 文書 Y のスコア は 75 となる。
スコア総和方式	論理積 および 論理和 (AND 条件お よび OR 条件)	演算対象 (検索条件) の文書内 でのそれぞ れの出現数 の合計値を 基にスコア が付けられ る。	「A および B が含まれる文 書」、または 「A または B が含まれる文 書」の検索で ヒットした文 書内の A と B の出現数の合 計を基にスコ アが付けられ る。	図 2-4 で「A およ び B が含まれる 文書」または 「A または B が含まれる文 書」を検索 すると、文書 X のスコアは 100、 文書 Y のスコア は 100 となる。

(凡例)

A : 検索条件 A

B : 検索条件 B

注 ここでの算出方法は、検索条件式の構造に重みが付けられていないことが前提です。構造に重みが付けられている場合は、算出結果が異なります。

## (8) 重み (重要度) を付けた検索

『インターネット』および『検索システム』を含む文書を検索したい。ただし、『検索システム』に重みを付けて検索したい」という場合があります。

重み (重要度) は複数の検索タームを指定する場合、重要度を高くしたい方の単語に付けます。上記の例で、「検索システム」に「10」、「インターネット」に「1」の割合で重みを付けたとします。この場合、「検索システム」は「インターネット」の 10 倍の重みが付けられます。ランキング検索の指定時、重みを付けた検索タームはより高いスコアが付けられます。

## 2. 検索の概要

ただし、論理積を条件にして重みを付けた場合、重み付けとは関係なくスコアが付与される場合があります（表 2-3 参照）。したがって、論理積を条件にして重みを付ける場合は、極端な配分にならないように注意してください。

重みは検索タームのほか、検索対象にする構造にも付けられます。

参照例：

検索タームに重みを付けて指定した例：表 2-6 の項番 9

構造に重みを付けて指定した例：表 2-7 の項番 3

### （9）検索結果のソート

（1）～（8）で説明した検索方法で文書を取得する場合、文書 ID の昇順にソートされます。ただし、ランキング検索の SORT オプション指定時は、文書 ID の昇順にソートされません。

検索結果のソートは、この文書 ID の順番を、関連情報ファイルに登録されているテキストデータをキーにして、昇順または降順に並び替える機能です。この機能を使うと、例えば、検索対象ファイルに書籍情報が登録されていて、関連情報ファイルにその書籍の発行日が登録されている場合、「作者名で検索し、その書籍の発行順で取得する」といった検索が実行できます。

参照例：

検索結果のソートの例：表 2-10 の項番 1～2

### （10）文章の概念を手がかりとした検索

例えば、漠然としたイメージだけがあって適切なキーワードが思いつかない場合があります。概念検索はこのような場合に利用します。

概念検索では、ユーザーが任意に指定した文章や文字列を手がかりにして、その条件と似た概念を持つ文書を検索します。例えば、「『近年、環境保護に関する対策に力を入れている自治体が増えている』という概念を持つ文書を探す」などの検索ができます。

概念検索の検索条件に指定する文章を、種文章といいます。概念検索では、初めに種文章を特徴づける単語が、種文章から抽出されます。この単語を特徴タームといいます。次に、抽出された特徴タームの中から、種文章の概念を表すタームが選出されます。ここで選出された特徴タームを検索用特徴タームといいます。この検索用特徴タームが実際の検索に使用されます。

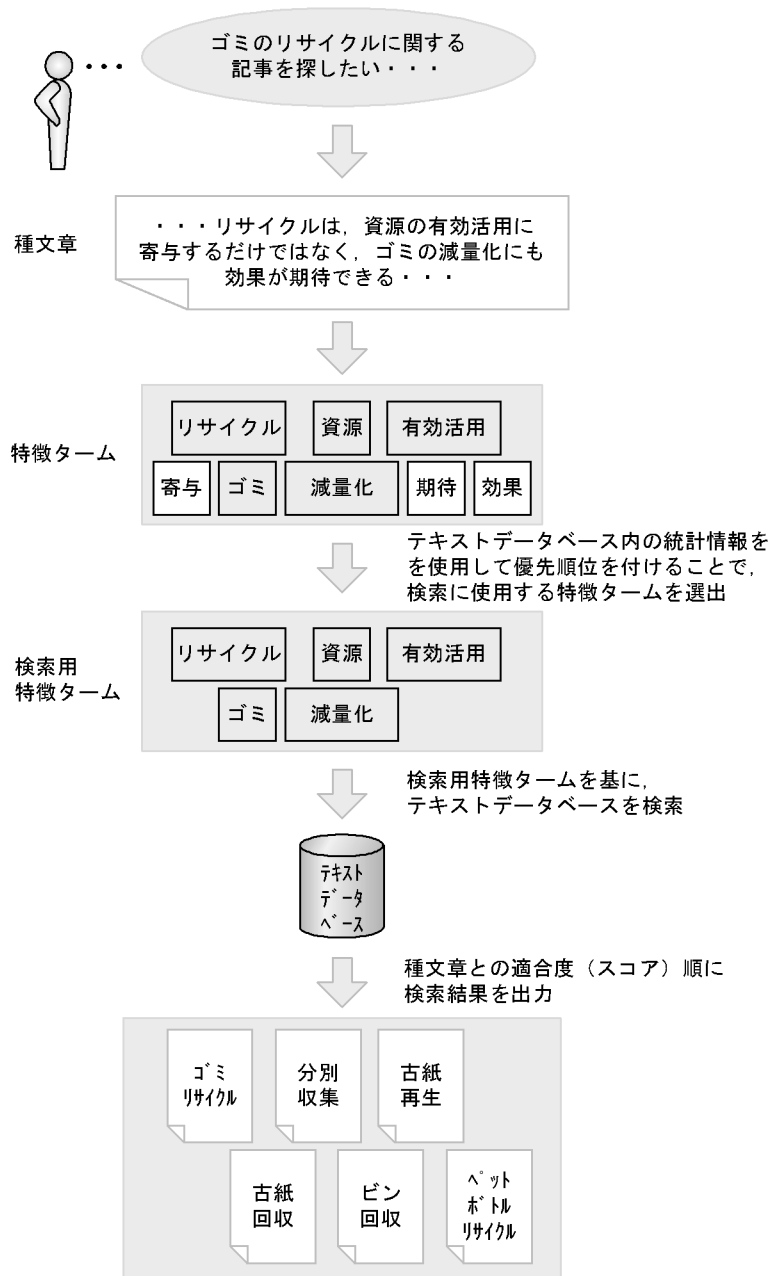
検索用特徴タームの抽出には、テキストデータベース内の統計情報を使用します。検索用特徴タームは、次の優先順位に従って種文章から抽出されます。

- 種文章に多く出現する特徴タームほど優先順位が高くなる
- テキストデータベース中に存在する数が少ない特徴タームほど優先順位が高くなる

上記の優先順位に従って抽出された検索用特徴タームを基に、テキストデータベース内の文書が検索されます。

概念検索の概要を次の図に示します。

図 2-5 概念検索の概要



参照例：

概念検索の事項例：表 2-11 の項番 1 ~ 4

Bibliotheca2 TS では、次の概念検索を実行できます。

## 2. 検索の概要

### (a) 種文章を複数指定する検索

検索条件には種文章を複数個指定できます。例えば、「近年、環境保護に関する対策に力を入れている自治体が増えている」という文章と「リサイクルは、資源の有効活用に寄与するだけでなく、ゴミの減量化にも効果が期待できる」という文章を種文章として指定し、それぞれの概念の和で表される仮想的な種文章に似た概念を持つ文書を検索できます。

### (b) 同義語・異表記展開検索

概念検索での同義語展開検索では、同義語辞書での定義に基づいて、検索用特徴タームの同義語が自動的に展開されます。概念検索での異表記展開検索では、Bibliotheca2 TSでのルールに基づいて、自動的に検索用特徴タームの異表記が展開されます。展開された同義語・異表記は、検索用特徴タームとあわせて検索条件となります。

### (c) 論理演算検索

概念検索での論理演算検索は、例えば、『近年、環境保護に関する対策に力を入れている自治体が増えている』という文章と似た概念を持つ文書を検索した結果と、『リサイクルは、資源の有効活用に寄与するだけでなく、ゴミの減量化にも効果が期待できる』という文章と似た概念を持つ文書を検索した結果の両方に含まれる文書を検索できます。この例では、検索条件同士の論理積（AND 条件）を検索条件にしています。

また、論理和（OR 条件）を検索の条件にして、『近年、環境保護に関する対策に力を入れている自治体が増えている』という文章と似た概念を持つ文書を検索した結果と、『リサイクルは、資源の有効活用に寄与するだけでなく、ゴミの減量化にも効果が期待できる』という文章と似た概念を持つ文書を検索した結果のどちらかに含まれる文書を検索することもできます。

### (d) 構造名を指定した検索

構造化文書を格納した表の列を検索対象とする場合は、その表の構造化文書を格納した列に定義された文書の構造を検索条件として利用できます。例えば、「文書・章・節・項」という構造を持つ文書を格納する列があるとします。この列に対しては、「節の中に『近年、環境保護に関する対策に力を入れている自治体が増えている』という文章と似た概念を持つ文書」のように、構造を条件にして検索できます。

### (e) スコア検索

概念検索の結果文書に、種文章の概念に対する適合度から算出した得点（スコア）を付けます。適合度が高い文書ほど得点が高くなります。

スコア算出時には、例えば「種文章を 100 点として、相対的な値に得点を調整する（スコアを正規化する）」、「一定のスコア以上の文書だけを検索結果とする」などの指定もできます。

### (f) 検索用特徴ターム出力

検索条件として指定した種文章から抽出される、検索用特徴タームを出力します。概念

検索ではこれらの特徴タームから、検索結果の妥当性を判断します。

(g) 検索タームによる絞り込み検索

概念検索で文書を抽出しておき、その抽出結果に対して検索タームで絞り込むという、概念検索と全文検索の長所を利用した検索を実現できます。この機能を利用することで、概念検索結果が散漫になることを防止できます。

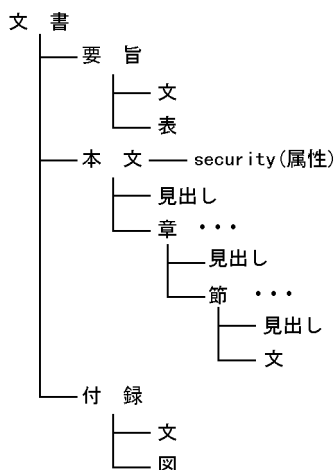
## 2.2 いろいろな検索と条件式の例題

この節では、Bibliotheca2 TS でのいろいろな検索方法を例題とあわせて説明します。

### (1) 例題の使い方

例題は、図 2-6 および図 2-7 に示す構造の文書を格納したテキストデータベースを想定しています。

図 2-6 例題に使用するテキストデータベース（検索対象ファイル）に格納された文書の構造



検索対象ファイル名：FindFile

図 2-7 例題に使用するテキストデータベース（関連情報ファイル）に格納された文字列の構造

次のフォーマットで文書の作成日時が登録されています。

フォーマット：YYYY/MM/DD hh:mm:ss

YYYY：西暦4けた  
 MM：月2けた（01～12）  
 DD：日2けた（01～31）  
 hh：時間2けた（00～23）  
 mm：分2けた（00～59）  
 ss：秒2けた（00～59）

関連情報ファイル名：InfoFile

例題は、サーチコマンド MFIND, MSORT, または MCONCEPT に指定する条件式の形式で説明しています。



サーチコマンド MFIND, MSORT, または MCONCEPT の詳細については、「4. サーチコマンドの文法」を参照してください。

例題の見方について説明します。

#### 項番

「2.1 Bibliotheca2 TS での検索」での例題の参照先と一致しています。

#### 検索の種類

「2.1 Bibliotheca2 TS での検索」で説明した検索の種類を挙げています。

#### 検索の内容

検索条件の内容を、次の凡例に基づいて説明しています。

< >: この括弧で囲まれた文字列は、テキストデータベースの構造名を示します。

「 」: この括弧で囲まれた文字列は、検索タームを示します。

#### 記述例

サーチコマンド MFIND, MSORT, または MCONCEPT に指定する条件式の指定例を示します。なお、検索対象ファイル名は「FindFile」、関連情報ファイル名は「InfoFile」としています。

## (2) 条件式 (MFIND) の例題

検索 (サーチコマンド MFIND) に使用する条件式の例題を、表 2-6 ~ 表 2-9 に示します。なお、表 2-6 ~ 表 2-9 に示す例題での検索対象ファイル名は「FindFile」としています。

表 2-6 検索条件式を使用した例題

項番	検索の種類	検索の内容	記述例
1	検索タームを 1 語指定した検索	「最新技術」を含む文書を検索する。	FindFile:eq('最新技術')
2	構造名を指定した検索	<本文>に「最新技術」を含む文書を検索する。	FindFile<文書.本文>: eq('最新技術')
3	検索タームにワイルドカードを指定した検索	任意の 2 文字で始まる「システム」という語を含む文書を検索する。	FindFile:eq('??システム')
4		<本文>に「M」で始まり「Y」で終わる文字列を含む文書を検索する。	FindFile<文書.本文>: eq('M*Y')
5		「How?」を含む文書を検索する。	FindFile:eq('How?')

## 2. 検索の概要

項番	検索の種類	検索の内容	記述例
6	検索タームに除外文字を指定した検索	任意の2文字で始まる「システム」という語を含む文書の中で、「検索システム」を含む文書を除外して検索する。	FindFile:eq('^検索システム')
7	検索ターム同士の論理積	「最新」および「情報」の両方を含む文書を検索する。	FindFile: fand('最新','情報')
8	ランキング検索	「最新」および「情報」の両方を含む文書を検索する。ただし、それぞれの検索タームが多く含まれる文書から検索結果を取り出す。	FindFile: fand('最新','情報'), RANK COUNTS SORT
9	検索タームに重みを付けた検索	「最新」および「情報」の両方を含む文書を検索する。ただし「最新」に5倍の重みを付けて検索する。	FindFile:for ( '最新' [5], '情報' ), RANK COUNTS SORT
10	検索ターム同士の論理和	「最新」または「情報」を含む文書を検索する。	FindFile: for('最新','情報')
11	近傍条件検索	「最新」と「情報」の間の文字数が20文字ちょうどの文字列を含む文書を検索する。	FindFile: c=20('最新','情報')
12		「最新」と「情報」の間の文字数が20文字ちょうどで、かつ「最新」と「情報」の順序で現れる文字列を含む文書を検索する。	FindFile: C=20('最新','情報')
13		「最新」と「情報」の間の文字数が20文字以下の文字列を含む文書を検索する。	FindFile: c<=20('最新','情報')
14		「最新」と「情報」の間の文字数が20文字以上の文字列を含む文書を検索する。	FindFile: c>=20('最新','情報')
15	同義語・異表記展開検索	「PC」を含む文書を検索する。ただし、同義語展開、アルファベット異表記展開および全角・半角異表記展開した結果も条件にして検索する。	FindFile:eq('PC') , EXPANS Uae

項番	検索の種類	検索の内容	記述例
16	繰り返し構造に対する検索	<章>に「最新」と「情報」を含む文書を検索する。ただし「最新」と「情報」は同じ章にある文書を検索する。	FindFile <文書.本文.章>: fand('最新','情報')
17	タグ属性を条件にする検索	<本文>の属性"security"の値が"secret"である文書を検索する。	FindFile<文書.本文>: security: eq('secret')

表 2-7 複合条件式を使用した例題

項番	検索の種類	検索の内容	記述例
1	検索条件式同士の論理積	「PC」を含む文書、および「最新」と「情報」の間の文字数が20文字ちょうどの文字列を含む文書の論理積を求めて検索する。	AND (FindFile:eq('PC') ), FindFile:c=20('最新', '情報'))
2	複数の構造を指定した検索	<要旨>に「インターネット」を含み、かつ<本文>に「PC」を含む文書を検索する。	AND (FindFile<文書.要旨>: eq('インターネット' '), FindFile<文書.本文>: eq('PC'))
3	構造に重みを付けた検索	<要旨>に「インターネット」を含むか、または<本文>に「PC」を含む文書を検索する。ただし、構造名<本文>に10倍の重みを付けて検索する。	OR (FindFile <文書.要旨>: eq('インターネット' '), FindFile<文書.本文>[10]: eq('PC')), RANK COUNTS SORT
4	検索条件式同士の論理和	「PC」を含む文書、および「最新」と「情報」の間の文字数が20文字ちょうどの文字列を含む文書の論理和を求めて検索する。	OR (FindFile:eq('PC' ), FindFile:c=20 ( '最新', '情報 '))
5	検索結果同士の論理積	IDが10、および12の検索結果集合に含まれる文書IDごとの論理積を求めて検索する。	AND(SET(10),SET(12))

## 2. 検索の概要

項番	検索の種類	検索の内容	記述例
6	検索結果の否定	「社会」という単語が存在しない文書を検索する。	NOT (FindFile:eq('社会'))
7		見出しに「社会」が存在し、かつ本文に「政治」が存在する文書以外の文書を検索する。	NOT (AND(FindFile<章・見出し>:eq('社会'), FindFile<章・本文>:eq('政治')))
8		見出しに「社会」が存在するか、または本文に政治が存在しない文書を検索する。	OR (FindFile<章・見出し>:eq('社会'), NOT(FindFile<章・本文>:eq('政治')))

表 2-8 特定構造検索条件式を使用した例題

項番	検索の種類	検索の内容	記述例
1	特定の構造に対する検索	<章>の<見出し>に「PC」を含み、かつ<節>の<見出し>に「インターネット」を含む文書を検索する。ただし、二つの検索タームは、同じ章に存在することとする。	SAND (FindFile<文書・本文・章>:FindFile<文書・本文・章・見出し>:eq('PC'), FindFile<文書・本文・章・見出し>:eq('インターネット'))

表 2-9 範囲指定検索条件式を使用した例題

項番	検索の種類	検索の内容	記述例
1	範囲指定検索	西暦 2000 年の範囲を検索する。	InfoFile:rg('2000/01/01 00:00:00,2000/12/31 23:59:59')
2		西暦 2001 年以降の範囲を検索する。	InfoFile:rg('2001/01/01 00:00:00,')
3		西暦 2000 年以前の範囲を検索する。	InfoFile:rg(',2000/12/31 23:59:59')

### (3) 条件式 (MSORT) の例題

ソートに使用する条件式の例題を表 2-10 に示します。なお、例題での関連情報ファイル名は「InfoFile」としてしています。

表 2-10 検索結果のソートの条件式を使用した例題

項番	検索の種類	検索の内容	記述例
1	ソート	サーチコマンド MFIND の検索結果集 合 ID5 の結果を、作成 日時の古い順にソート する。	MSORT SET(5),InfoFile
2		サーチコマンド MFIND の検索結果集 合 ID10 の結果を、作 成日時の新しい順に ソートする。	MSORT SET(10),InfoFile, DESC

### (4) 条件式 (MCONCEPT) の例題

概念検索 (サーチコマンド MCONCEPT) に使用する条件式の例題を、次の表に示します。なお、例題での検索対象ファイル名は「FindFile」としてしています。

表 2-11 概念検索の条件式を使用した例題

項番	検索の種類	検索の内容	記述例
1	概念検索	種文章を一つ指定して 検索する。	MCONCEPT FindFile: eq('種文 章')
2		種文章を一つ指定して ランキング検索し、検 索結果をソートする。	MCONCEPT RANK SORT FindFile: eq('種文 章')
3		種文章を二つ指定して ランキング検索し、検 索結果をソートする。	MCONCEPT RANK SORT FindFile: eq('種文 章1','種文章2')
4		見出しが種文章 1 に該 当するか、または本文 が種文章 2 に該当しな い文書を検索する。	MCONCEPT OR (FindFile <章. 見出し>:eq('種文章 1'),NOT(FindFile <章.本文>:eq('種文 章2')))

注 : 上段はサーチコマンド、下段は条件式を示しています。



# 3

## 関数の文法

この章では、Development Kit または Server Library が提供する関数の文法について説明します。

---

3.1 関数の概要

---

3.2 使用するファイル

---

3.3 関数の発行順序

---

3.4 UAP を作成する上での注意

---

3.5 Development Kit または Server Library で使用する関数

---

## 3.1 関数の概要

Development Kit または Server Library は、テキストデータベースを検索する UAP を C 言語で作成するために使用します。作成した UAP は、TS サーバーと接続して、クライアントプログラムとして使用できます。

Development Kit または Server Library で使用する関数を、次の表に示します。

表 3-1 Development Kit または Server Library で使用する関数

関数名	機能
TS2F_cancelcmd	サーチコマンドを中断する
TS2F_canceldat	サーチコマンドの実行結果を破棄する
TS2F_cmd	サーチコマンドを実行する
TS2F_cmd_DATA	入力データ付きサーチコマンドを実行する
TS2F_con	TS サーバーとの通信セッションを確立する
TS2F_discon	TS サーバーとの通信セッションを切断する
TS2F_geterror	詳細エラー情報を取得する
TS2F_gettermpos	テキストデータ中の検索タームのヒット位置を取得する
TS2F_mlooknv	サーチコマンド MLOOK の実行結果を文字列に変換する
TS2F_rcvdat	サーチコマンドの実行結果を取得して終了を確認する
TS2F_sesopt	セッションオプションを操作する
TS2F_sndcmd	サーチコマンドの実行を TS サーバーに要求する
TS2F_sndcmd_DATA	入力データ付きサーチコマンドの実行を TS サーバーに要求する
TS2F_well_formed_xmlcpl	簡易 XML 文書解析機能を使用して、整形 XML 文書を全文検索用データに変換する
TS2F_well_formed_xmlcpl_free	TS2F_well_formed_xmlcpl 関数で取得した全文検索用データ格納領域を解放する



## 3.2 使用するファイル

UAP の作成に使用するファイルおよび各ファイルの格納位置を、OS ごとに示します。

### 3.2.1 OS が Windows の場合

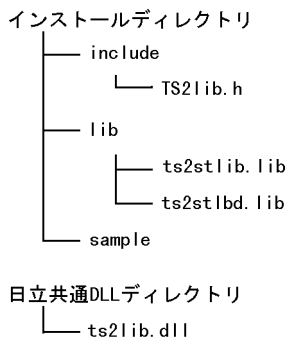
OS が Windows の場合に使用するファイルを、次の表に示します。

表 3-2 使用するファイル (Windows の場合)

ファイルの種類	ファイル名	説明
ヘッダファイル	TS2lib.h	UAP にインクルードして使用するファイル
DLL ファイル	ts2lib.dll	UAP を実行するときに使用するファイル
スタティックライブラリ	ts2stlib.lib ts2stlbd.lib	UAP のリンケージ時に使用するファイル

注 UAP が使用するランタイムライブラリが「マルチスレッド (DLL)」の場合、ts2stlib.lib をリンケージしてください。ランタイムライブラリが「マルチスレッド (DLL, デバッグ)」の場合は、ts2stlbd.lib をリンケージしてください。

図 3-1 ファイルの格納位置



### 3.2.2 OS が UNIX の場合

OS が UNIX の場合に使用するファイルを、次の表に示します。

表 3-3 使用するファイル (UNIX の場合)

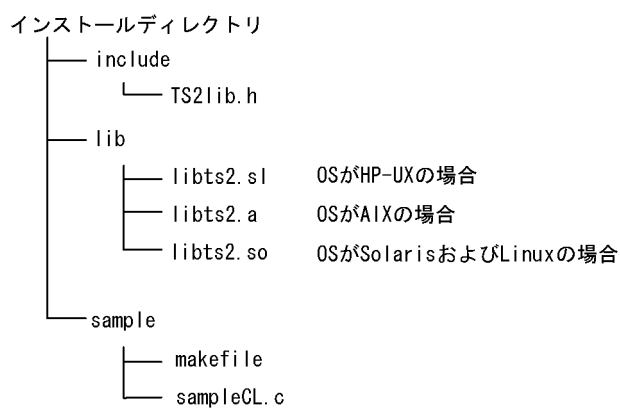
ファイルの種類	ファイル名	説明
ヘッダファイル	TS2lib.h	UAP にインクルードして使用するファイル

### 3. 関数の文法

ファイルの種類	ファイル名	説明
シェアドライブラリ	<ul style="list-style-type: none"><li>OS が HP-UX の場合 libts2.sl</li><li>OS が AIX の場合 libts2.a</li><li>OS が Linux または Solaris の場合 libts2.so</li></ul>	UAP を実行するときに 使用するファイル

ファイルの格納位置は使用する OS によって異なります。OS 別のファイル構成を下記の図に示します。

図 3-2 ファイルの格納位置



インストールディレクトリを次に示します。

`/opt/Bib2TS/TS2/library`

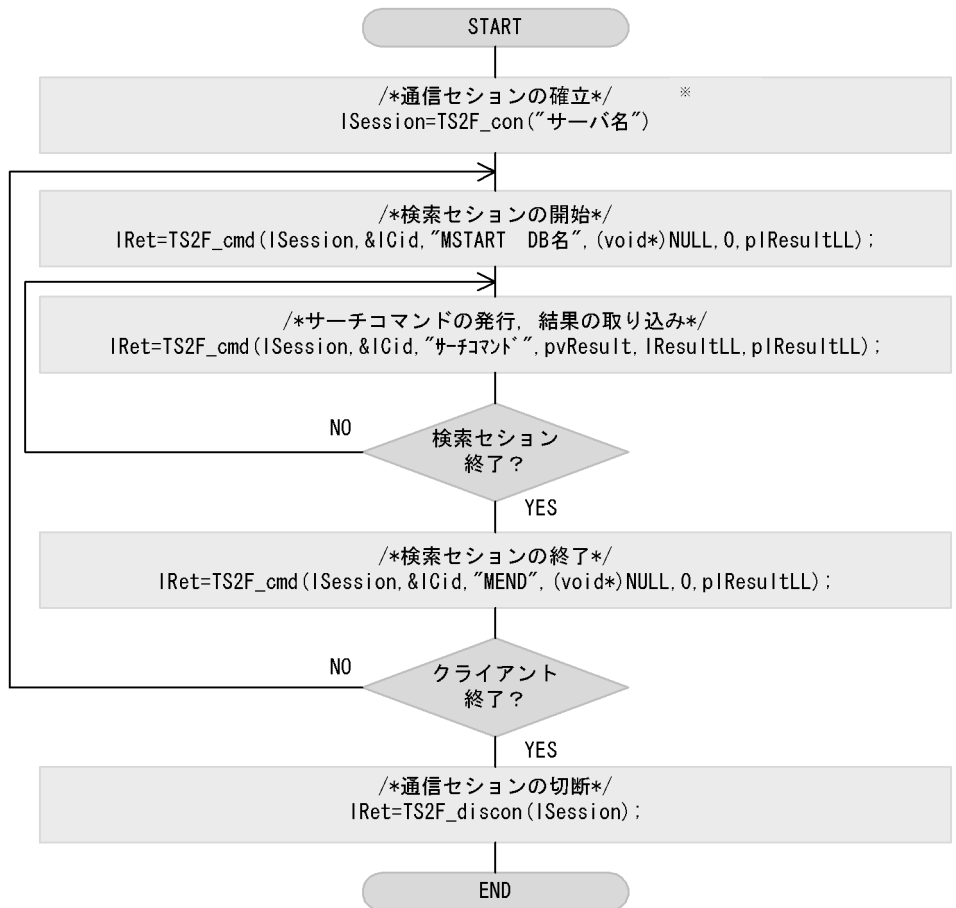
## 3.3 関数の発行順序

ここでは、関数の発行順序を示します。

### 3.3.1 TS2F\_cmd 関数でサーチコマンドを処理する

TS2F\_cmd 関数は、サーチコマンドを発行して結果の取り込みまで一度に行う関数です。TS2F\_cmd 関数を使用してサーチコマンドを実行する順序を、次の図に示します。

図 3-3 TS2F\_cmd 関数の処理手順



注※ TS2F\_con関数を実行する前に、環境変数TSLANGの設定が必要です。詳細は、「付録A.3 文字コード種別の環境設定」を参照してください。

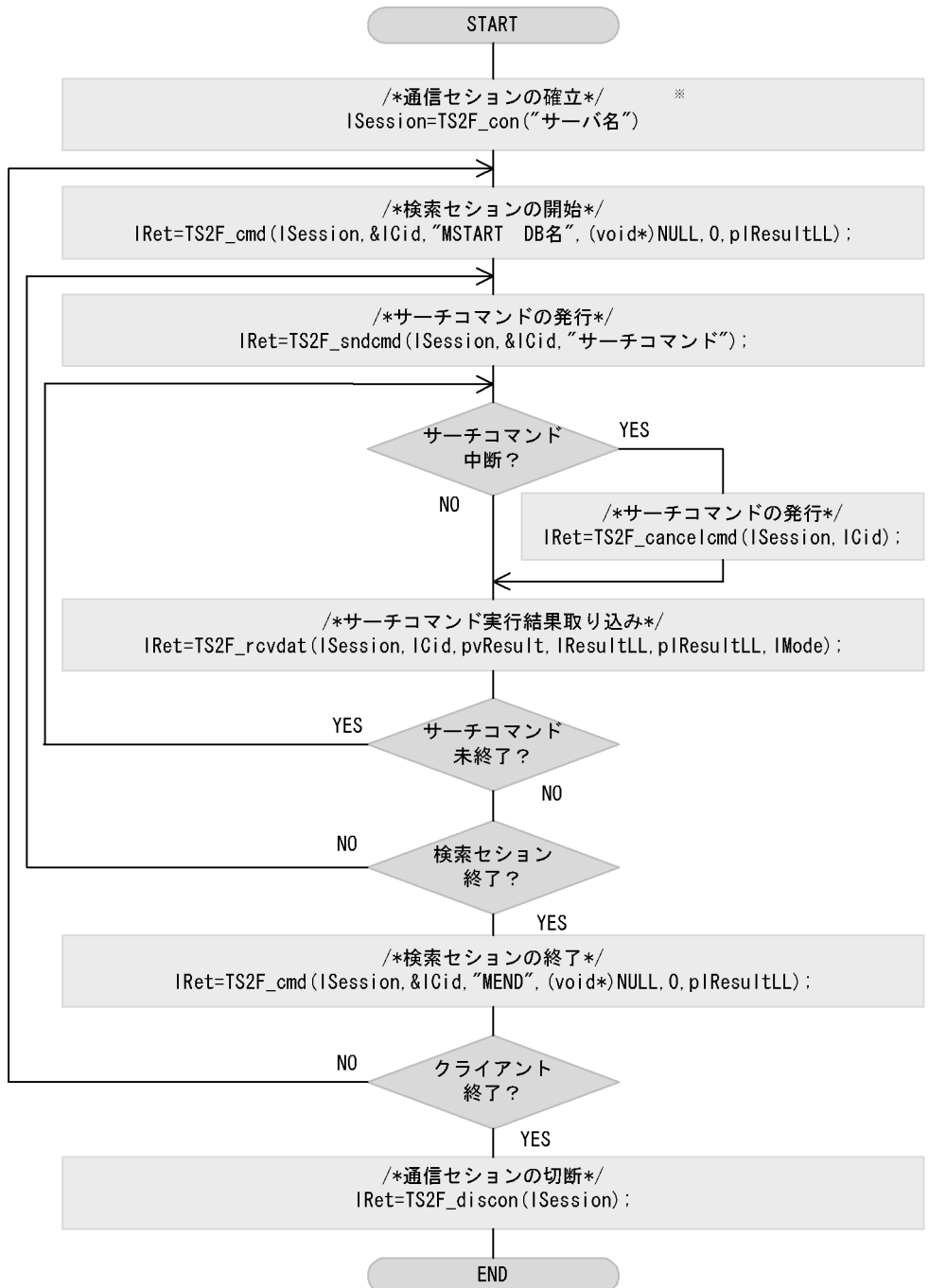
### 3.3.2 TS2F\_sndcmd 関数および TS2F\_rcvdat 関数でサーチコマンドを処理する

TS2F\_sndcmd 関数は、サーチコマンドを発行する関数です。TS2F\_rcvdat 関数は、サーチコマンドの実行結果を取り込む関数です。TS2F\_sndcmd 関数、および TS2F\_rcvdat 関数は組み合わせて使用します。

TS2F\_cmd 関数と異なり、TS2F\_sndcmd 関数を使うと、サーチコマンドを発行したあとに制御が一度戻るため、サーチコマンドをキャンセルできます。

TS2F\_sndcmd 関数、および TS2F\_rcvdat 関数でサーチコマンドを実行する順序を、次の図に示します。

図 3-4 TS2F\_sndcmd 関数および TS2F\_rcvdat 関数の処理手順



注※ TS2F\_con関数を発行する前に、環境変数TSLANGの設定が必要です。詳細は、「付録A.3 文字コード種別の環境設定」を参照してください。

## 3.4 UAP を作成する上での注意

---

UAP を作成する上での注意を OS ごとに説明します。

### 3.4.1 OS が Windows の場合

UAP をリンケージするときは、リンケージライブラリ `ts2stlib.lib` を使用してリンケージしてください。これによって、UAP の実行時に DLL ファイル `ts2lib.dll` がロードされます。DLL ファイル `ts2lib.dll` が存在しない場合は、テキストサーチ機能を利用できないことを示すエラーコードが返されます。

UAP を作成するときの Visual C++ のプロジェクト設定方法は、Visual C++ のバージョンにより異なります。例として Visual C++ 5.0 での手順を次に示します。

1. Visual C++ を起動する
2. プロジェクトを作成する
  - メニューから「ファイル」 - 「新規作成」を選択します
  - 「プロジェクト」タグを選択します
  - 作成する UAP のタイプを選択し、プロジェクト名や位置などを入力します
3. インクルードヘッダ位置を設定する
  - メニューから「プロジェクト」 - 「設定」を選択します
  - 設定の対象に Debug と Release の両方を選択した状態で「C/C++」タグを選択します
  - カテゴリから「プリプロセッサ」を選択します
  - インクルードファイルのパスに、「Development Kit または Server Library をインストールしたディレクトリ `¥include`」を追加します
4. C ランタイムライブラリのデバッグ版の設定をする
  - メニューから「プロジェクト」 - 「設定」を選択します
  - 設定の対象に「Debug」を選択した状態で「C/C++」タグを選択します
  - カテゴリから「コード生成」を選択します
  - 使用するランタイムライブラリを「マルチスレッド (DLL, デバッグ)」に設定します
  - 上記の状態から「リンク」タグを選択します
  - カテゴリから「一般」を選択します
  - オブジェクト/ライブラリモジュールに、「Development Kit または Server Library をインストールしたディレクトリ `¥lib¥ts2stlibd.lib`」を追加します
5. C ランタイムライブラリのリリース版の設定をする
  - メニューから「プロジェクト」 - 「設定」を選択します
  - 設定の対象に「Release」を選択した状態で「C/C++」タグを選択します
  - カテゴリから「コード生成」を選択します
  - 使用するランタイムライブラリを「マルチスレッド (DLL)」に設定します

- 上記の状態から「リンク」タグを選択します
  - カテゴリから「一般」を選択します
  - オブジェクト/ライブラリモジュールに、「Development Kit または Server Library をインストールしたディレクトリ ¥lib¥ts2stlib.lib」を追加します
6. プロジェクトにファイルを追加する
    - UAP のソースファイルをプロジェクトに追加します
  7. プログラムをビルドする
    - リンケージの際に、次のようなメッセージが出力される場合、プロジェクトのコードの設定およびリンケージ時に使用した Development Kit または Server Library のスタティックライブラリを確認してください。

「LINK:warning LNK4098:defaultlib"xxxx" は他のライブラリの使用と競合していません; ~」

### 3.4.2 OS が UNIX の場合

Bibliotheca2 TS 上で管理されるテーブルなどの情報は、すべて Bibliotheca2 TS 用のインクルードファイルである TS2lib.h 内で宣言されています。したがって、UAP 上で Server Library の関数を使用するときには、次の内容を定義してください。

```
#include "TS2lib.h"
```

コンパイルには、次のオプションを指定してください。

#### OS が HP-UX の場合

- -I/opt/Bib2TS/TS2/library/include
- -L/opt/Bib2TS/TS2/library/lib
- -lts2

#### OS が AIX の場合

- -qchars=signed -qalias=noansi
- -I/opt/Bib2TS/TS2/library/include
- -L/opt/Bib2TS/TS2/library/lib
- -lts2

#### OS が Solaris の場合

- -I/opt/Bib2TS/TS2/library/include
- -L/opt/Bib2TS/TS2/library/lib
- -misalign -lts2 -lsocket -lpthread -lnsl

上記のオプションを指定した場合、シェアドライブラリがリンケージされます。作成したプログラムを実行するときは、LD\_LIBRARY\_PATH 環境変数に「opt/

### 3. 関数の文法

BibTS/TS2/Library/lib」を設定してください。

OS が Linux の場合

- `-I/opt/Bib2TS/TS2/library/include`
- `-Wl, -rpath, /opt/BibTS/TS2/Library/lib`
- `-L/opt/Bib2TS/TS2/library/lib`
- `-lts2`



## 3.5 Development Kit または Server Library で使用する関数

---

この節では、Development Kit または Server Library で使用する関数について説明します。

### 3.5.1 説明する項目

各関数で説明する項目を次に示します。

#### (1) 形式

関数の記述形式を示します。

#### (2) 機能

関数の機能について説明しています。

#### (3) 引数

関数の引数を示します。後ろに《in》が付いている引数は、値を入力する引数を意味します。後ろに《out》が付いている引数は、関数によって値が設定される引数を意味します。

#### (4) 戻り値

関数の戻り値を示します。

#### (5) 注意事項

関数を使用する上での注意事項を示します。

#### (6) 指定例

関数の指定例を示します。

### 3.5.2 TS2F\_cancelcmd (サーチコマンドの中断)

#### (1) 形式

```
long TS2F_cancelcmd(long lSession,long lCommand)
```

#### (2) 機能

サーチコマンドの実行を中断します。

### 3. 関数の文法

#### (3) 引数

ISession 《in》

TS2F\_con 関数で返されたセッション識別子を指定します。

ICommand 《in》

中断するサーチコマンドのコマンド情報 ID を指定します。コマンド情報 ID は、TS2F\_sndcmd 関数から返されます。

#### (4) 戻り値

0

正常終了

負の整数値

エラー番号を示します。詳細については、「付録 E 関数のエラーステータス」を参照してください。

#### (5) 注意事項

サーチコマンドが中断されたかどうかを確認するには、TS2F\_rcvdat 関数でサーチコマンドの実行結果を取得して確認してください。エラーステータス -130010 (エラーマクロ名 TS2DERL\_CCLED\_CID) が返されれば、サーチコマンドは中断されています。エラーステータスの詳細については、「付録 E 関数のエラーステータス」を参照してください。

#### (6) 指定例

```
#include "TS2lib.h"

long IWsess; /* セッション識別子 */
long IWrtm; /* 戻り値受け取り用 */
char *pcWdata; /* サーチコマンド実行結果受け取り用 */
long IWcommand; /* コマンド識別子受け取り用 */
long IWLL; /* データ受け取りエリアサイズ */
long IWrslt; /* 実行結果サイズ受け取り用 */

/* IWLLに受け取りエリアサイズを設定する */
/* pcWdataへ長さIWLL分の領域を割り当てる */
/* セッション接続処理 (TS2F_con) */
/* サーチコマンド実行要求 (TS2F_sndcmd参照) */

IWrtm = TS2F_cancelcmd (IWsess, IWcommand);
if (IWrtm == 0) {
    /* TS2DL_WAITモードでTS2F_rcvdatを発行してサーチコマンド実行結果を取得 */
} else {
    /* サーチコマンドキャンセル時の処理 */
}
```

### 3.5.3 TS2F\_canceldat (サーチコマンドの実行結果の破棄)

#### (1) 形式

```
long TS2F_canceldat(long lSession, long lCommand)
```

#### (2) 機能

サーチコマンドの実行結果を破棄します。

#### (3) 引数

lSession 《in》

TS2F\_con 関数で返されたセッション識別子を指定します。

lCommand 《in》

実行結果を破棄するサーチコマンドのコマンド情報 ID を指定します。コマンド情報 ID は、次の各関数から返されます。

- TS2F\_cmd
- TS2F\_sndcmd

#### (4) 戻り値

0

正常終了

負の整数値

エラー番号を示します。詳細については、「付録 E 関数のエラーステータス」を参照してください。

#### (5) 注意事項

- TS2F\_canceldat 関数では、TS2F\_cmd 関数または TS2F\_rcvdat 関数で実行結果不足 (エラーステータス -130016 (エラーマクロ名 TS2DERL\_LESS\_BUFSIZ)) になったサーチコマンドの実行結果だけを破棄できます。エラーステータスの詳細については、「付録 E 関数のエラーステータス」を参照してください。
- サーチコマンドの実行結果が破棄されたかどうかは、TS2F\_rcvdat 関数でサーチコマンドの実行結果を取得して確認してください。エラーステータス -130010 (エラーマクロ名 TS2DERL\_CCLED\_CID) が返されれば、実行結果は破棄されています。エラーステータスの詳細については、「付録 E 関数のエラーステータス」を参照してください。

### 3. 関数の文法

#### (6) 指定例

```
#include "TS2lib.h"

long lWsess: /* セッション識別子 */
long lWrtn: /* 戻り値受け取り用 */
char *pcWdata: /* サーチコマンド実行結果受け取り用 */
long lWcommand: /* コマンド識別子受け取り用 */
long lWLL: /* データ受け取りエリアサイズ */
long lWrsIt: /* 実行結果サイズ受け取り用 */

/* lWLLに受け取りエリアサイズを設定する */
/* pcWdataへ長さlWLL分の領域を割り当てる */
/* セッション接続処理(TS2F_con) */
/* サーチコマンドの実行または結果受け取り時にTS2DERL_LESS_BUFSIZエラー発生 */

lWrtn = TS2F_canceldat(lWsess, lWcommand);
if(lWrtn == 0) {
    /*TS2DL_WAITモートでTS2F_rcvdatを発行しサーチコマンド実行を終了させる */
    /* サーチコマンド実行結果破棄成功時の処理 */
} else {
    /* サーチコマンド実行結果破棄エラー時の処理 */
}
```

### 3.5.4 TS2F\_cmd (サーチコマンドの実行)

#### (1) 形式

```
long TS2F_cmd(long lSession, long *plCommand,
              char *strCommand, void *pvResult,
              long lResultLL, long *plResultLL)
```

#### (2) 機能

サーチコマンドを TS サーバーへ発行し、実行結果を引数 pvResult に取り込みます。さらに、実行結果のサイズ(バイト数)を引数 plResultLL で示す領域に設定します。

#### (3) 引数

lSession 《in》

TS2F\_con 関数で返されたセッション識別子を指定します。

plCommand 《out》

コマンド情報 ID を受け取る領域のポインタを指定します。

strCommand 《in》

実行するサーチコマンドを指定します。

pvResult 《out》

サーチコマンドの実行結果を受け取る領域のポインタを指定します。

lResultLL 《in》

サーチコマンドの実行結果を受け取る領域のサイズを指定します。

plResultLL 《out》

サーチコマンドの実行結果のサイズを受け取る領域のポインタを指定します。

#### (4) 戻り値

0

正常終了

負の整数値

エラー番号を示します。詳細については、「付録 E 関数のエラーステータス」を参照してください。

#### (5) 注意事項

- 引数 lResultLL に設定した実行結果領域のサイズより実行結果のサイズが大きい場合、サーチコマンドの実行結果が受け取れないで、エラーステータス -130016 (エラーマクロ名 TS2DERL\_LESS\_BUFSIZ) が戻り値として返されます。この場合は次のように対応してください。

対応 1 : 実行結果が必要な場合

### 3. 関数の文法

1. 引数 plResultLL に設定されている実行結果領域を大きくする  
このとき、すべての実行結果を受け取るだけの領域を確保してください。
2. TS2F\_rcvdat 関数の引数 lMode に TS2DL\_RETRY を指定して実行結果を受け取る

対応2：実行結果が不要な場合

TS2F\_canceldat 関数を使用し、実行結果を破棄する

実行結果を受け取るかまたは破棄しないと、次のコマンドを実行できません。

エラーステータスの詳細については、「付録 E 関数のエラーステータス」を参照してください。

- 引数 strCommand に指定できるサーチコマンドの最大長は、終端の NULL 文字を含めて 1,024 バイトです。
- 1メガバイトを超える実行結果は、サーチコマンド側で1メガバイトに格納できるように調整されます。  
サーチコマンド MDISPLAY の実行結果は、1メガバイトを超える場合があります。  
サーチコマンド MDISPLAY は、文書の何バイトから何バイトまでを取得する、という指定ができます。このため、1メガバイトを超える文書を取得する場合は、次に示すように取得してください。
- 1メガバイトを超える文書を取得する場合は、一つの文書を分割して取得してください。
- 同時に複数の文書を取得する場合は、一度サーチコマンド MDISPLAY を実行した時点でどこまで取得できたかを確認して、再度取得する文書を変更して実行してください。

## (6) 指定例

```

#include "TS2lib.h"

long lWsess; /* ショウ識別子 */
long lWrtn; /* 戻り値受け取り用 */
char *pcWdata; /* サーチコマンド実行結果受け取り用 */
long lWcommand; /* コマンド識別子受け取り用 */
long lWLL; /* データ受け取りエリアサイズ */
long lWrslt; /* 実行結果サイズ受け取り用 */

/* lWLLに受け取りエリアサイズを設定する */
/* pcWdataへ長さlWLL分の領域を割り当てる */
/* ショウ接続処理(TS2F_con) */

lWrtn = TS2F_cmd(lWsess, &lWcommand, "MDBINF DB", pcWdata, lWLL, &lWrslt);
if(lWrtn == 0) {
    /* サーチコマンド実行成功時の処理 */
} else {
    if(TS2DERL_LESS_BUFSIZ == lWrtn) {
        if(結果の受け取り) {
            /* 結果受け取り処理へ(TS2F_rovdatのTS2DL_RETRYモード参照) */
        } else {
            /* 実行結果破棄へ(TS2F_canceldat参照) */
        }
    } else {
        /* サーチコマンドエラー時の処理 */
    }
}
}

```

### 3.5.5 TS2F\_cmd\_DATA (入力データ付きサーチコマンドの実行)

#### (1) 形式

```
long TS2F_cmd_DATA(long lSession,    long *plCommand,  
                  char *strCommand,void *pvIndata,  
                  long lIndataLL,   void *pvResult,  
                  long lResultLL,   long *plResultLL)
```

#### (2) 機能

入力データが必要なサーチコマンドを TS2 サーバーへ発行し、実行結果を引数 pvResult に取り込みます。さらに、実行結果のサイズ(バイト数)を引数 plResultLL に設定します。

#### (3) 引数

lSession 《in》

TS2F\_con 関数で返されたセッション識別子を指定します。

plCommand 《out》

コマンド情報 ID を受け取る領域のポインタを指定します。

strCommand 《in》

実行するサーチコマンドを指定します。

pvIndata 《out》

入力データの領域のポインタを指定します。

lIndataLL 《in》

入力データ長を指定します。

pvResult 《out》

サーチコマンドの実行結果を受け取る領域のポインタを指定します。

lResultLL 《in》

サーチコマンドの実行結果を受け取る領域のサイズを指定します。

plResultLL 《out》

サーチコマンドの実行結果のサイズを受け取る領域のポインタを指定します。

#### (4) 戻り値

0

正常終了

負の整数値

エラー番号を示します。詳細については、「付録 E 関数のエラーステータス」を参



照してください。

### (5) 注意事項

- 引数 plResultLL に設定した実行結果領域のサイズより、実行結果のサイズが大きい場合、エラーステータス -130016 (エラーマクロ名 TS2DERL\_LESS\_BUFSIZ) が戻り値として返されます。この場合は次のように対応してください。

対応 1 : 実行結果が必要な場合

1. 引数 plResultLL に設定されている実行結果領域を大きくする  
このとき、すべての実行結果を受け取るだけの領域を確保してください。
2. TS2F\_rcvdat 関数の引数 Mode に TS2DL\_RETRY を指定して実行結果を受け取る

対応 2 : 実行結果が不要な場合

TS2F\_canceldat 関数を使用し、実行結果を破棄する

実行結果を受け取るかまたは破棄しないと、次のコマンドを実行できません。

エラーステータスの詳細については、「付録 E 関数のエラーステータス」を参照してください。

- 引数 strCommand に指定できるサーチコマンドの最大長は、終端の NULL 文字を含めて 1,024 バイトです。
- この関数に渡す入力データのサイズは、最大 1 メガバイトまでです。

### 3. 関数の文法

#### (6) 指定例

```
#include "TS2lib.h"

long lWsess;      /* ショウ識別子 */
long lWrtn;      /* 戻り値受け取り用 */
char *pcWdata;   /* サーチコマンド実行結果受け取り用 */
long lWcommand;  /* コマンド識別子受け取り用 */
long lWLL;       /* データ受け取りエリアサイズ */
long lWrslt;     /* 実行結果サイズ受け取り用 */
char *pcWldata;  /* 入力データ */
long lWInLL;     /* 入力データサイズ */

/* lWLLに受け取りエリアサイズを設定する */
/* pcWdataへ長さlWLL分の領域を割り当てる */
/* ショウ接続処理(TS2F_con) */
/* MSTARTを実行 */
/* pcWldataへ領域を割り当て、入力データを設定する */
/* lWInLLに入力データサイズを設定する */

lWrtn = TS2F_cmd_DATA(lWsess, &lWcommand, "MDATAMAKE FindFile", pcWldata, lWInLL,
pcWdata, lWLL, &lWrslt);
if(lWrtn == 0) {
    /* サーチコマンド実行成功時の処理 */
} else {
    if(TS2DERL_LESS_BUFSIZ == lWrtn) {
        if(結果の受け取り) {
            /* 結果受け取り処理へ(TS2F_rcvdatのTS2DL_RETRYト参照) */
        } else {
            /* 実行結果破棄へ(TS2F_canceldat参照) */
        }
    } else {
        /* サーチコマンドエラー時の処理 */
    }
}
}
```

### 3.5.6 TS2F\_con (通信セッションの確立)

#### (1) 形式

```
long TS2F_con(char *strServer)
```

#### (2) 機能

引数 strServer に指定された TS サーバーとの間にセッションを確立します。

#### (3) 引数

strServer 《in》

接続する TS サーバーのサーバー名を指定します。

#### (4) 戻り値

0 以上の整数値

セッション識別子を示します。セッション識別子は、サーチコマンドを発行する場合などに使用します。

負の整数値

エラー番号を示します。詳細については、「付録 E 関数のエラーステータス」を参照してください。

#### (5) 注意事項

通信ソケットの使用方法は環境変数で設定できます。詳細は「付録 A.4 通信ソケットの環境設定」を参照してください。

#### (6) 指定例

```
#include "TS2lib.h"

long lWsess; /* セッション識別子 */

lWsess = TS2F_con("TSSERVER"); /* "TSSERVER" という名称のサーバーに接続 */
if(lWsess < 0) {
    /* 接続失敗時の処理 */
}
/* 接続成功時の処理 */
```

### 3.5.7 TS2F\_discon (通信セッションの切断)

#### (1) 形式

```
long TS2F_discon(long lSession)
```

#### (2) 機能

TS2F\_con 関数で接続した通信セッションを切断します。

#### (3) 引数

lSession 《in》

TS2F\_con 関数で返されたセッション識別子を指定します。

#### (4) 戻り値

0

正常終了

負の整数値

エラー番号を示します。詳細については、「付録 E 関数のエラーステータス」を参照してください。

#### (5) 注意事項

なし。

#### (6) 指定例

```
#include "TS2lib.h"

long lWsess; /* セッション識別子 */
long lWrtn; /* 戻り値受け取り用 */

/* セッション接続処理(TS2F_con) */

lWrtn = TS2F_discon(lWsess); /* TS2F_conで接続したセッションを解放する */
if(lWrtn == 0) {
    /* 解放成功時の処理 */
} else {
    /* 解放失敗時の処理 */
}
```

### 3.5.8 TS2F\_geterror ( 詳細エラー情報の取得 )

#### ( 1 ) 形式

```
long TS2F_geterror(char *pcMsg, long lMsgLL,
                  long *plDetail)
```

#### ( 2 ) 機能

関数発行時の詳細エラー情報を取得します。

サーバー側でエラーが発生し、サーバー側が詳細メッセージを返している場合に取得できます。

#### ( 3 ) 引数

pcMsg 《out》

詳細メッセージのテキストを受け取る領域のポインタを指定します。

lMsgLL 《in》

詳細メッセージのテキストを受け取る領域のサイズを指定します。

plDetail 《out》

詳細エラーコードを受け取る領域のポインタを指定します。

#### ( 4 ) 戻り値

エラー発生種別が返されます。

0

エラーは発生していないか、またはエラー情報が存在しません。

TS2DERL\_SERVER

サーバーでエラーが発生しました。

TS2DERL\_CLIENT

クライアントでエラーが発生しました。

#### ( 5 ) 注意事項

- この関数では、直前に発行した関数のエラー情報を取得できます。
- 引数 lMsgLL に設定した領域を超えるメッセージテキストが設定された場合、指定された領域に格納できるだけのテキストがコピーされます。なお、メッセージテキストの末端には、NULL 文字 '\0' が設定されます。メッセージテキストに漢字が含まれる場合、漢字コードの途中でメッセージが中断されるときがあります。

### 3. 関数の文法

#### (6) 指定例

```
#include "TS2lib.h"

char *pcWMsg;
long lWrtn;
long lWMsgLL;
long lWDetail;

/* lWMsgLLに1024を設定して、pcWMsgにlWMsgLL分の領域を割り当てる
(1キロバイトまでのメッセージを受け取る場合) */
/* サーバメントの実行時、結果を受け取る時にエラー発生 */

lWrtn = TS2F_geterror(pcWMsg, lWMsgLL, &lWDetail);
switch(lWrtn) {
    case 0:
        /* エラー情報なし */
        break;
    case TS2DERL_SERVER:
        /* サーバでエラー */
        break;
    case TS2DERL_CLIENT:
        /* クライアントでエラー */
        break;
    default:
        /* TS2F_geterrorリターン不正 */
        break;
}
```

### 3.5.9 TS2F\_gettermpos ( 検索タームのヒット位置の取得 )

#### (1) 形式

```
long TS2F_gettermpos(void *pvText, long lTextLL,
                    void *pvMlook,
                    TS2T_HITPOS *ptHitpos,
                    long lEntry, long *plPosCount)
```

#### ヒット情報構造体

```
typedef struct TS2T_HITPOS {
    long lHitPos;
    long lHitLL;
} TS2T_HITPOS ;
```

#### (2) 機能

サーチコマンド MLOOK OUTMODE ALL での実行結果を基にテキストデータを検索し、指定された検索タームのヒット位置を取得します。

#### (3) 引数

##### pvText 《in》

ヒット位置を取得するテキストデータのポインタを指定します。

##### lTextLL 《in》

テキストデータの長さを指定します。

##### pvMlook 《in》

サーチコマンド MLOOK OUTMODE ALL の実行結果を受け取る領域を指定します。

##### ptHitpos 《out》

ヒット情報を受け取る TS2T\_HITPOS 領域のポインタを指定します。

##### lEntry 《in》

引数 ptHitpos に指定した領域に受け取れるヒット情報の件数を指定します。

##### plPosCount 《out》

引数 ptHitpos に設定された、ヒット情報の件数を受け取る領域のポインタを指定します。

#### ヒット情報構造体

##### lHitPos 《out》

- ヒットした位置が、テキストの先頭からのバイト数で設定されます。

##### lHitLL 《out》

- ヒットした文字列の長さがバイト数で設定されます。

### 3. 関数の文法

#### (4) 戻り値

0

正常終了

負の整数値

エラー番号を示します。詳細については、「付録 E 関数のエラーステータス」を参照してください。

#### (5) 注意事項

なし。

#### (6) 指定例

```
#include "TS2lib.h"

char *pcWmlook;      /* MLOOK OUTMODE ALL結果受け取り領域 */
char *pcWtext;      /* ヒット位置情報を取得したいテキスト */
long lWtext;        /* テキスト長 */
TS2T_HITPOS *ptWhit; /* ヒット位置情報受け取り領域 */
long lWCount;       /* ヒット位置情報受け取り回数 */
long lWhit;         /* ヒット位置情報数受け取り用 */
long lWrtn;         /* 戻り値受け取り用 */

/* IWLLに 1024を設定して、pIWdataにIWLL*sizeof(TS2T_HITPOS)分の領域を割り当てる
(1024までの結果受け取り時) */
/* OUTMODE ALL指定のMLOOKコマンドを実行して、実行結果をpcWmlookに受け取る */
/* ヒット位置情報を取得するテキストをpcWtextに設定し、長さをlWtextに設定 */

lWrtn = TS2F_gettermpos (pcWtext, lWtext, pcWmlook, ptWhit, lWCount, &lWhit);
if (lWrtn != 0) {
    /* TS2F_gettermpos実行エラー */
}
```



### 3.5.10 TS2F\_mlookcnv (サーチコマンド MLOOK の実行結果の変換)

#### (1) 形式

```
long TS2F_mlookcnv(void *pvMlook, long *plTermArea,
                  long lAreaLL, long *plResultLL)
```

#### (2) 機能

サーチコマンド MLOOK のオプションに OUTMODE ALL が指定されている場合、実行結果から異表記展開された文字列を生成します。

異表記展開された文字列は、引数 plTermArea に次の形式で設定されます。



#### (3) 引数

pvMlook 《in》

サーチコマンド MLOOK OUTMODE ALL の実行結果を受け取る領域を指すポインタを指定します。

plTermArea 《out》

異表記展開の結果の文字列を受け取る領域のポインタを指定します。

lAreaLL 《in》

異表記展開の結果の文字列を格納する領域のサイズを指定します。

plResultLL 《out》

異表記展開の結果のサイズを受け取る領域のポインタを指定します。

#### (4) 戻り値

0

正常終了

負の整数値

エラー番号を示します。詳細については、「付録 E 関数のエラーステータス」を参照してください。

### 3. 関数の文法

#### (5) 注意事項

なし。

#### (6) 指定例

```
#include "TS2lib.h"

char *pcWmlook; /* MLOOK OUTMODE ALL結果受け取り領域 */
long *pIWdata; /* mlookcnv実行結果受け取り領域 */
long IWLL; /* mlookcnv実行結果受け取り領域サイズ */
long IWrsIt; /* mlookcnv実行結果サイズ */
long IWrtN; /* 戻り値受け取り用 */
long IWCount; /* 展開ターム表示用カウンタ */
char *pcWterm; /* 展開ターム */

/* IWLLに(1024*1024)を設定し、pIWdataにIWLL分の領域を割り当てる
(1メガバイトまでの結果受け取り時) */
/* OUTMODE ALL指定のサチコメント MLOOKを実行し、実行結果をpcWmlookに受け取る */

IWrtN = TS2F_mlookcnv(pcWmlook, pIWdata, IWLL, &IWrsIt);
if(IWrtN != 0) {
    /* TS2F_mlookcnv実行エラー */
}

/* 展開ターム表示 */
printf("展開ターム数:%6d\n", pIWdata[0]); /* 展開ターム数表示 */
pcWterm = (char *)(&pIWdata[1]); /* 先頭のタームアドレスへ位置づけ */
for(IWCount=0; IWCount< pIWdata[0]; IWCount++){
    printf("TERM:%s\n", pcWterm); /* 展開タームの表示 */
    pcWterm += strlen(pcWterm) + 1;
}
```

### 3.5.11 TS2F\_rcvdat (サーチコマンドの実行結果の取得および終了の確認)

#### (1) 形式

```
long TS2F_rcvdat(long lSession ,long lCommand,
                void *pvResult,
                long lResultLL, long *plResultLL,
                long lMode)
```

#### (2) 機能

TS2F\_sndcmd 関数で発行したサーチコマンドの実行結果を受け取ります。さらに、TS2F\_cmd 関数または TS2F\_rcvdat 関数で実行結果不足エラーステータス -130016 (エラーマクロ名 TS2DERL\_LESS\_BUFSIZ) となったサーチコマンドの実行結果を受け取ります。

エラーステータスの詳細については、「付録 E 関数のエラーステータス」を参照してください。

#### (3) 引数

lSession 《in》

TS2F\_con 関数で返されたセッション識別子を指定します。

lCommand 《in》

TS2F\_cmd 関数または TS2F\_sndcmd 関数が返したコマンド情報 ID を指定します。

pvResult 《out》

サーチコマンドの実行結果を受け取る領域のポインタを指定します。

lResultLL 《in》

サーチコマンドの実行結果を受け取る領域のサイズを指定します。

plResultLL 《out》

サーチコマンドの実行結果のサイズを受け取る領域のポインタを指定します。

lMode 《in》

実行結果を受け取る時のモードを指定します。指定できる値は次のとおりです。

- TS2DL\_WAIT  
サーチコマンドの実行が終了してから実行結果を受け取ります。
- TS2DL\_NOWAIT  
サーチコマンドの実行が終了している場合、実行結果を受け取ります。サーチコマンドの実行が終了していない場合、エラーステータス -130035 (エラーマクロ名 TS2DERL\_NOTEND) が戻り値として返されます。エラーステータスの詳細については、「付録 E 関数のエラーステータス」を参照してください。
- TS2DL\_RETRY

### 3. 関数の文法

実行結果不足 ( エラーステータス -130016 ( エラーマクロ名 TS2DERL\_LESS\_BUFSIZ )) となった実行結果を受け取ります。エラーステータスの詳細については、「付録 E 関数のエラーステータス」を参照してください。

#### (4) 戻り値

0

正常終了

負の整数値

エラー番号を示します。詳細については、「付録 E 関数のエラーステータス」を参照してください。

#### (5) 注意事項

1 メガバイトを超える実行結果はサーチコマンド側で1メガバイトに格納できるように調整されます。

サーチコマンド MDISPLAY の実行結果は、1メガバイトを超える場合があります。サーチコマンド MDISPLAY は、文書の何バイトから何バイトまでを取得する、という指定ができます。このため、1メガバイトを超える文書を取得する場合は、次に示すように取得してください。

- 1メガバイトを超える文書を取得する場合は、一つの文書を分割して取得してください。
- 同時に複数の文書を取得する場合は、一度サーチコマンド MDISPLAY を実行した時点でどこまで取得できたかを確認して、再度取得する文書を変更して実行してください。

#### (6) 指定例

サーチコマンドの終了まで待ち、実行結果を受け取る場合 ( TS2DL\_WAIT モード )

```

#include "TS2lib.h"

long lWsess; /* ション識別子 */
long lWrtn; /* 戻り値受け取り用 */
char *pcWdata; /* サーチコマンド実行結果受け取り用 */
long lWcommand; /* コマンド識別子受け取り用 */
long lWLL; /* データ受け取りエリアサイズ */
long lWrslt; /* 実行結果サイズ受け取り用 */

/* lWLLに受け取りエリアサイズを設定する */
/* pcWdataへ長さlWLL分の領域を割り当てる */
/* ション接続処理(TS2F_con) */
/* サーチコマンド実行要求(TS2F_sndcmd参照) */

lWrtn =TS2F_rcvdat(lWsess, lWcommand, pcWdata, lWLL, &lWrslt, TS2DL_WAIT);
if(lWrtn == 0) {
    /* サーチコマンド実行結果取得成功時の処理 */
} else {
    if(TS2DERL_LESS_BUFSIZ == lWrtn) {
        if(結果の受け取り) {
            /* 結果受け取り処理へ(TS2F_rcvdatのTS2DL_RETRYト参照) */
        } else {
            /* 実行結果破棄へ(詳細はTS2F_canceldat参照) */
        }
    } else {
        /* サーチコマンドエラー時の処理 */
    }
}
}

```

サーチコマンドの終了を待たないで、実行結果を受け取る場合 (TS2DL\_NOWAIT  
モード)

### 3. 関数の文法

```
#include "TS2lib.h"

long lWsess; /* セッション識別子 */
long lWrtn; /* 戻り値受け取り用 */
char *pcWdata; /* サーチコマンド実行結果受け取り用 */
long lWcommand; /* コマンド識別子受け取り用 */
long lWLL; /* データ受け取りエリアサイズ */
long lWrslt; /* 実行結果サイズ受け取り用 */

/* lWLLに受け取りエリアサイズを設定する */
/* pcWdataへ長さlWLL分の領域を割り当てる */
/* セッション接続処理(TS2F_con) */
/* サーチコマンド実行要求(TS2F_sndcmd参照) */

while(1) {
    lWrtn = TS2F_rcvdat(lWsess, lWcommand, pcWdata, lWLL, &lWrslt, TS2DL_NOWAIT);
    if(lWrtn == 0) {
        /* 実行結果受け取り成功時の処理 */
        break; /* whileを抜ける */
    } else {
        if(TS2DERL_LESS_BUFSIZ == lWrtn) {
            if(結果の受け取り) {
                /* 結果受け取り処理へ(TS2F_rcvdatのTS2DL_RETRYモード参照) */
            } else {
                /* 実行結果破棄へ(TS2F_canceldat参照) */
            }
            break; /* whileを抜ける */
        } if(TS2DERL_NOTEND != lWrtn) { /* サーチコマンド実行中以外 */
            /* サーチコマンドエラー時の処理 */
            break; /* whileを抜ける */
        }
    }
}
/* キャンセルの判定処理など */
if(キャンセル要) {
    /* サーチコマンドキャンセル処理(TS2F_cancelcmd参照) */
    /* TS2DL_WAITモードでTS2F_rcvdatを発行しサーチコマンド実行結果を受け取る */
    break; /* whileを抜ける */
}
}
```

実行結果の不足時、実行結果を再度受け取る場合 (TS2DL\_RETRY モード)

```

#include "TS2lib.h"

long lWsess; /* セッション識別子 */
long lWrtn; /* 戻り値受け取り用 */
char *pcWdata; /* サブコマンド実行結果受け取り用 */
long lWcommand; /* コマンド識別子受け取り用 */
long lWLL; /* テータ受け取りエリアサイズ */
long lWrslit; /* 実行結果サイズ受け取り用 */

/* lWLLに受け取りエリアサイズを設定する */
/* pcWdataへ長さlWLL分の領域を割り当てる */
/* セッション接続処理(TS2F_con) */
/* サブコマンドの実行または結果の受け取りでTS2DERL_LESS_BUFSIZエラー */
/* lWLLにサブコマンド実行結果サイズ(lWrslit)以上の値を設定する */
/* pcWdataの領域をlWLL分再割り当てる */

lWrtn = TS2F_rcvdat(lWsess, lWcommand, pcWdata, lWLL, &lWrslit, TS2DL_RETRY);
if(lWrtn == 0) {
    /* サブコマンド実行結果取得成功時の処理 */
} else {
    /* サブコマンド実行結果取得エラー時の処理 */
}

```

## 3.5.12 TS2F\_sesopt (セッションオプションの操作)

### (1) 機能

IMode に指定したセッションオプションに従って、セッションオプションを操作します。

### (2) 形式

```
long TS2F_sesopt(long lSession, long lMode,  
TS2T_SESOPT *ptSesopt, unsigned long ulSesoptsiz)
```

セッションオプション構造体

```
typedef struct TS2T_SESOPT {  
    unsigned long ulSendDataType;  
} TS2F_SESOPT;
```

### (3) 引数

lSession 《in》

TS2F\_con 関数で返されたセッション識別子を指定します。

IMode 《in》

セッションオプションの操作モードを指定します。指定できる値は次のとおりです。

- TS2DL\_GET  
セッションオプションを指定します。
- TS2DL\_SET  
セッションオプションを取得します。

ptSesopt 《in》

セッションオプションテーブルのポインタを指定します。

lSesoptsiz 《in》

セッションオプションテーブルのサイズを指定します。sizeof (TS2T\_SESOPT) を指定してください。

セッションオプション構造体

ulSendDataType 《in》

- TS2DL\_SDT\_MDTMK : M DATAMAKE 用入力データ (デフォルト)
- TS2DL\_SDT\_BINARY : バイナリデータ

### (4) 戻り値

0

正常終了

負の整数値

エラー番号を示します。詳細については、「付録 E 関数のエラーステータス」を参照してください。



## (5) 注意事項

複数のスレッドから同一セッションに対してセッションオプションを設定した場合、最後に設定したオプションが有効となります。

## (6) 指定例

```
include "TS2lib.h"

long lWsess;          /* セッション識別子 */
TS2T_SESOPT tWsesopt; /* セッションオプション */

/* セッション接続処理 (TS2F_con) */

/* 送信データ種別をバインディングに設定 */
tWsesopt.ulSendDataType = TS2DL_SDT_BINARY;

lWrtn = TS2F_sesopt(lWsess, TS2DL_SET, &tWsesopt, sizeof(TS2T_SESOPT));
if(lWrtn == 0) {
    /* セッションオプション操作成功時の処理 */
} else {
    /* セッションオプション操作失敗時の処理 */
}
```

### 3.5.13 TS2F\_sndcmd (サーチコマンドの実行の要求)

#### (1) 形式

```
long TS2F_sndcmd(long lSession, long *plCommand,  
                char *strCommand)
```

#### (2) 機能

サーチコマンドを TS サーバーへ発行します。

サーチコマンドの終了の確認および実行結果の受け取りには、TS2F\_rcvdat 関数を使用します。実行中のサーチコマンドの中断には TS2F\_cancelcmd 関数を使用します。

#### (3) 引数

lSession 《in》

TS2F\_con 関数で返されたセッション識別子を指定します。

plCommand 《out》

コマンド情報 ID を受け取る領域のポインタを指定します。

strCommand 《in》

実行するサーチコマンドを指定します。

#### (4) 戻り値

0

正常終了

負の整数値

エラー番号を示します。詳細については、「付録 E 関数のエラーステータス」を参照してください。

#### (5) 注意事項

引数 strCommand に設定できるサーチコマンドの最大長は、終端の NULL 文字を含めて 1,024 バイトです。

## (6) 指定例

```
#include "TS2lib.h"

long lWsess: /* セッション識別子 */
long lWrtn: /* 戻り値受け取り用 */
long lWcommand: /* コマンド識別子受け取り用 */

/* セッション接続処理(TS2F_con) */

lWrtn = TS2F_sndcmd(lWsess, &lWcommand, "MDBINF DB");
if(lWrtn == 0) {
    /* 結果受け取り処理へ(TS2F_rcvdatのTS2DL_WAITモードまたはTS2DL_NOWAITモード参照)*/
} else {
    /* サーブコマンド実行要求エラー時の処理 */
}
```

### 3.5.14 TS2F\_sndcmd\_DATA (入力データ付きサーチコマンドの実行の要求)

#### (1) 形式

```
long TS2F_sndcmd_DATA(long lSession, long *plCommand,  
                      char *strCommand, void *pvIndata,  
                      long lIndataLL)
```

#### (2) 機能

入力データが必要なサーチコマンドを TS サーバーへ発行します。

サーチコマンドの終了の確認および実行結果の受け取りには、TS2F\_revdat 関数を使用します。実行中のサーチコマンドの中断には TS2F\_cancelcmd 関数を使用します。

#### (3) 引数

lSession 《in》

TS2F\_con 関数で返されたセッション識別子を指定します。

plCommand 《out》

コマンド情報 ID を受け取る領域のポインタを指定します。

strCommand 《in》

実行するサーチコマンドを指定します。

pvIndata 《out》

入力データ領域のポインタを指定します。

lIndataLL 《in》

入力データ長を指定します。

#### (4) 戻り値

0

正常終了

負の整数値

エラー番号を示します。詳細については、「付録 E 関数のエラーステータス」を参照してください。

#### (5) 注意事項

- 引数 strCommand に設定できるサーチコマンドの最大長は、終端の NULL 文字を含み 1,024 バイトです。
- この関数に渡す入力データのサイズは、最大 1 メガバイトまでです。

## (6) 指定例

```

#include "TS2lib.h"

long lWsess: /* セッション識別子 */
long lWrtn: /* 戻り値受け取り用 */
long lWcommand: /* コマンド識別子受け取り用 */
char *pcWldata: /* 入力データ */
long lWinLL: /* 入力データサイズ */

/* セッション接続処理 (TS2F_con) */
/* MSTARTを実行 */
/* pcWldataへ領域を割り当て、入力データを設定する */
/* lWinLLに入力データサイズを設定する */

lWrtn = TS2F_sndcmd_DATA(lWsess, &lWcommand, "MDATAMAKE FindFile", pcWldata, lWinLL);
if(lWrtn == 0) {
    /* 結果受け取り処理へ (TS2F_rcvdatのTS2DL_WAITモードまたはTS2DL_NOWAITモード参照) */
} else {
    /* サーチコマンド実行要求エラー時の処理 */
}

```

### 3.5.15 TS2F\_well\_formed\_xmlcpl ( 整形 XML 文書の全文 検索データ変換 )

#### ( 1 ) 形式

```
long TS2F_well_formed_xmlcpl (
    TS2T_WELLFORMEDXMOPT *ptXmlOpt,
    TS2T_WELLFORMEDXMLDAT *ptXmlDat,
    void **ppvResult,
    long *plResultLL
)
```

#### XML 変換オプション構造体

```
typedef struct TS2T_WELLFORMEDXMOPT (
    long lCplmode;
    long lCharaspec;
    long lReserve[4];
) TS2T_WELLFORMEDXMOPT;
```

#### XML データ情報構造体

```
typedef struct TS2T_WELLFORMEDXMLDAT (
    long lEntryCnt;
    TS2T_WELLFORMEDXMLENTRY tXmlent[1];
) TS2T_WELLFORMEDXMLDAT;
```

#### XML データエントリ構造体

```
typedef struct TS2T_WELLFORMEDXMLENTRY (
    long lTextid;
    long lXmldataLL;
    unsigned char *strXmldata;
) TS2T_WELLFORMEDXMLENTRY;
```

#### ( 2 ) 機能

簡易 XML 文書解析機能を使用して、整形 XML 文書を全文検索用データ、または MDATAMAKE の入力データ形式に変換します。MDATAMAKE の入力データ形式に変換した場合、そのまま MDATAMAKE コマンドの入力として扱うことができます。

#### ( 3 ) 引数

ptXmlOpt 《in》

XML 変換オプションのポインタを指定します。

ptXmlDat 《in》

XML データ情報のポインタを指定します。

ppvResult 《out》

全文検索用データ変換結果格納領域の受け取り領域ポインタを指定します。

plResultLL 《out》

全文検索用データ変換結果サイズの受け取り領域ポインタを指定します。

## XML 変換オプション構造体

## ICplmode 《in》

- TS2D\_ESISB\_ORG

全文検索用データに変換する場合に指定します。

変換対象の XML 文書のデータサイズが 3MB を超える場合、戻り値がエラーとなり ppvResult , および plResultLL に結果を返しません。

- TS2D\_ESISB\_MDATAMAKE

MDATAMAKE の入力データ形式に変換する場合に指定します。

変換結果のサイズが 1MB を超える場合、戻り値がエラーとなり ppvResult , および plResultLL に結果を返しません。

## ICharaspec 《in》

- TS2D\_SPEC\_JIS

登録先データベース文字コード種別が utf-8 , または utf-8\_ucs4 の場合に指定します。

- TS2D\_SPEC\_MS

登録先データベース文字コード種別が utf-8-ms , または utf-8-ms\_ucs4 の場合に指定します。

## IReserve[4] 《in》

- すべて 0 を指定します。

## XML データ情報構造体

## IEntryCnt 《in》

- 1 を指定します。

## tXmlent 《in》

- XML データエントリ構造体です。

## XML データエントリ構造体

## ITextid 《in》

- MDATAMAKE の入力データ形式に変換する場合に設定する文書 ID を指定します。

## IXmldataLL 《in》

- strXmldata に指定した XML 文書データサイズを指定します。

## strXmldata 《in》

- XML 文書データ格納ポインタを指定します。

## (4) 戻り値

0

正常終了

負の整数値

### 3. 関数の文法

エラー番号を示します。詳細については、「付録 E 関数のエラーステータス」を参照してください。

#### (5) 注意事項

- 正常終了した場合、TS2F\_well\_formed\_xmlcpl 関数で確保した領域ポインタを ppvResult に設定します。設定したポインタを解放する場合は、TS2F\_well\_formed\_xmlcpl\_free 関数を使用してください。
- lTextid には、MDATAMAKE で指定できる値を指定してください。
- エラー情報の詳細は、TS2F\_geterror 関数で取得してください。

#### (a) XML 文書の制限

TS2F\_well\_formed\_xmlcpl 関数で変換できる XML 文書には、次の制限事項があります。

##### XML 文書の構造

複数の XML 文書を一つのテキストデータベースに登録する場合は、最上位タグ名称を同じにしてください。下位の構造は同じにする必要はありません。

なお、テキストデータベースの最上位タグ名称は、テキストデータベースを作成したときに指定した、構造検証用ファイルの最上位タグ名称です。

##### タグ名称

タグ名称に半角英小文字が含まれている場合は、半角英大文字として扱われます。大文字小文字の区別はされません。

##### タグのネスト数

タグのネスト数（深さ）は 100 を上限とします。ただし、属性が指定できるネスト数（深さ）は 99 が上限です。

##### XML 文書の仕様

妥当性制約の検証は行ないません。

##### XML 文書の形式

処理できるのは整形形式な XML 文書です。DTD、XML スキーマ、コメント、接頭辞、および処理命令が記述されている場合は、記述されていないものとして処理されます。

##### 使用できる実体

使用できるのは、次に示す定義済み実体だけです。これ以外の場合は、文字として処理されます。

&lt; &gt; &amp; &apos; &quot;

##### タグ名、属性名の長さ

タグ名、属性名の長さは、UTF-8 エンコード換算で 4,096 バイトまで指定できます。

##### XML 文書のエンコーディング属性

XML 文書の文字コードは、XML 宣言の encoding 属性（encoding="XXX" の XXX



の部分)に指定します。

指定できるエンコーディング属性は UTF-8, または US-ASCII です。そのほかのエンコーディング属性の場合はエラーとなります。また, 指定を省略した場合は UTF-8 を仮定します。

#### 正規化

空白類が XML 文書のテキストノードに含まれる場合は, 次に示す正規化を行います。空白類とは空白 (0x20), 水平タブ (0x09), 改行 (0x0A), 復帰 (0x0D) をいいます。ただし, xml:space="preserve" 属性を持つ要素の子孫のテキストノードは正規化されません。

- 先頭, および末尾の空白類を削除します。
- 連続する空白類を一つの空白 (0x20) に置き換えます。

#### (6) 指定例

```
#include "TS2lib.h"

long          lWtn;
TS2T_WELLFORMEDXMOPT  tXmlOpt;
TS2T_WELLFORMEDXMLDAT  tXmlDat;
void          *pvResult;
long          lResultLL;

/* XML変換オプションおよびXMLデータ情報を設定 */

lWtn = TS2F_well_formed_xmlcpl(&tXmlOpt, &tXmlDat, &pvResult, &lResultLL);
if(lRtn == 0) {
    /* データ変換成功時の処理 */

    /* データ変換結果格納領域の解放 */
    TS2F_well_formed_xmlcpl_free(pvResult);
} else {
    /* データ変換失敗時の処理 */
}
```

### 3.5.16 TS2F\_well\_formed\_xmlcpl\_free ( 整形 XML 文書の全文検索データ格納領域解放 )

#### ( 1 ) 形式

```
void TS2F_well_formed_xmlcpl_free( void *pvResult )
```

#### ( 2 ) 機能

TS2F\_well\_formed\_xmlcpl 関数で取得した全文検索用データ格納領域を解放します。

#### ( 3 ) 引数

pvResult

TS2F\_well\_formed\_xmlcpl 関数で取得した全文検索用データ格納領域ポインタを指定します。

#### ( 4 ) 戻り値

なし。

#### ( 5 ) 注意事項

TS2F\_well\_formed\_xmlcpl 関数で取得した全文検索用データ格納領域の解放以外に使用しないでください。

#### ( 6 ) 指定例

TS2F\_well\_formed\_xmlcpl 関数の指定例を参照してください。

# 4

## サーチコマンドの文法

この章では、TS サーバーが提供するサーチコマンドの文法について説明します。

---

4.1 サーチコマンドの概要

---

4.2 サーチコマンドの使用上の注意

---

4.3 サーチコマンド

---

## 4.1 サーチコマンドの概要

サーチコマンドは、テキストデータベースの検索や文書の取得に使用するコマンドです。Development Kit または Server Library の関数を使って作成した UAP に記述し、TS サーバーに対して発行します。UAP では、関数のパラメーターに指定して実行します。

サーチコマンドの一覧を、次の表に示します。

表 4-1 サーチコマンドの一覧

サーチコマンド名	機能
MCONCEPT	概念を指定してテキストデータベースを検索する
MDATAMAKE	テキストデータベースに文書を追加する
MDATARM	テキストデータベースの文書を削除する
MDBINF	テキストデータベースに関する情報を出力する
MDISPLAY	検索した結果の文書を取得する
MEND	検索したテキストデータベースに対して使用を終了する宣言をする
MFIND (全文検索)	検索条件を指定してテキストデータベースを検索する
MFIND (範囲指定検索)	検索範囲を指定してテキストデータベースを検索する
MINCREMENTAL	テキストデータベースのインクリメンタル (最適化) を実行する
MLOOK	検索タームの同義語・異表記展開の結果を出力する
MOPT	検索オプションを指定する
MOSTAT	検索オプションの設定されている状態を取得する
MSORT	検索結果をソートする
MSTART	検索するテキストデータベースに対して使用を開始する宣言をする
MSWEEP	検索結果集合を削除する

## 4.2 サーチコマンドの使用上の注意

---

サーチコマンドを使用する上での注意を示します。

サーチコマンドは、テキストデータベース名や検索タームなどユーザーが指定する値を除いて、半角の英数字および記号で指定してください。

サーチコマンドのオプションを指定する場合は、半角1文字以上の空白を指定してください。

(例)

正しい記述 : MFIND 本文 :eq(' 情報処理 ')

誤った記述 : MFIND 本文 :eq(' 情報処理 ')

サーチコマンド MSTART を実行してからサーチコマンド MEND を実行するまでの検索できる状態を、検索セッションといいます。サーチコマンド MSTART および MDBINF 以外のサーチコマンドは、サーチコマンド MSTART で検索セッションを開始してから実行してください。

## 4.3 サーチコマンド

---

この節では、サーチコマンドの文法について説明します。

### 4.3.1 説明する項目

サーチコマンドの文法は、次の形式に従って説明します。

なお、該当する説明項目がない場合は、記述を省略しています。

#### (1) 形式

サーチコマンドを記述する形式を示します。

#### (2) 機能

サーチコマンドの機能について説明しています。

#### (3) オプション

サーチコマンドのオプションの指定方法について説明しています。

#### (4) 注意事項

サーチコマンドを使用する上での注意事項を示します。

#### (5) 指定例

サーチコマンドの指定例を示します。

#### (6) 入力データ形式

サーチコマンド実行時に入力するデータのデータ形式について説明しています。

#### (7) 実行結果の出力形式

サーチコマンドが実行結果として値を返す場合のデータ形式について説明しています。

### 4.3.2 MCONCEPT (概念検索)

#### (1) 形式

MCONCEPT [概念検索オプション [, 概念検索オプション] ...]

概念検索オプション

ランキングオプション | 検索オプション | ターム指定オプション

- ランキングオプション

RANK [SORT]

- 検索オプション

{EXPANS {展開モード}} | {BASE {検索結果集合 ID}}

展開モード:{U}{k}{a}{h}{e}

- ターム指定オプション

TERM

## (2) 機能

指定した条件に従って、概念検索用データベースを検索します。

## (3) オプション

概念検索のオプションを次の形式で指定します。

ランキングオプション | 検索オプション | ターム指定オプション

- ランキングオプション

検索結果に重み（重要度）を付けて検索する場合に指定します。ランキングオプションを指定して検索すると、それまでのランキング検索の結果は消去されます。ランキングオプションは RANK {SORT} の形式で指定します。

SORT：検索結果を重みの降順にソートします。

- 検索オプション

検索時の展開モード、またはベースを指定します。検索オプションを省略した場合は、サーチコマンド MOPT で指定された値が仮定されます。

検索オプションには EXPANS 句、または BASE 句を指定します。

- EXPANS {展開モード}

検索タームの展開モードを指定します。展開モードを省略すると、同義語・異表記展開検索されません。次の 5 種類を組み合わせで指定します。

展開モード	意味
U	検索タームを同義語展開する場合に指定します。ただし、同義語辞書が設定されていない場合は、エラーになります。
k	検索タームかたかな異表記展開する場合に指定します。
a	検索タームをアルファベット異表記展開する場合に指定します。
h	検索タームを半角アルファベット異表記展開する場合に指定します。
e	検索タームを全角・半角異表記展開する場合に指定します。

- BASE {検索結果集合 ID}

検索時のベースを指定します。

{検索結果集合 ID} には、検索のベースにする検索結果集合の ID を指定します。ここで指定した検索結果集合に含まれる文書が検索の対象になります。

なお、0 を指定するとすべての文書が検索の対象になります。検索結果集合 ID を省略すると 0 が仮定されます。

- ターム指定オプション

種文章から特徴タームを抽出しないで、種文章自体を検索タームとして検索する

#### 4. サーチコマンドの文法

場合に指定します。ターム指定オプションは、次の形式で指定します。

```
TERM
```

#### (4) 注意事項

このコマンドを実行する前にサーチコマンド MSTART を実行して、使用するテキストデータベースの検索セッションを開始しておいてください。

送信データ種別がバイナリデータではない場合は、TS2F\_sesopt() で送信データ種別をバイナリデータに設定してから実行してください。

データベースの文字コード種別が utf-8 または utf-8-ms の場合、UTF-8 シフト JIS UTF-8 の順序で変換を行なったときに、可逆性のある文字を検索の対象とします。可逆性のない文字の場合は、無視します。

#### (5) 指定例

指定例を次に示します。例では上段がサーチコマンドを、下段が条件式を示します。

種文章を一つ指定して検索する場合

```
MCONCEPT  
ファイル: eq('種文章')
```

種文章を一つ指定し、ランキング検索する場合

```
MCONCEPT RANK  
ファイル: eq('種文章')
```

種文章を一つ指定してランキング検索し、検索結果をソートする場合

```
MCONCEPT RANK SORT  
ファイル: eq('種文章')
```

種文章を二つ指定して検索する場合

```
MCONCEPT  
ファイル: eq('種文章1', '種文章2')
```

種文章を二つ指定し、ランキング検索する場合

```
MCONCEPT RANK  
ファイル: eq('種文章1', '種文章2')
```

種文章を二つ指定してランキング検索し、検索結果をソートする場合

```
MCONCEPT RANK SORT  
ファイル: eq('種文章1', '種文章2')
```

かたかな異表記展開を指定して検索する場合

```
MCONCEPT EXPANS k  
ファイル: eq('種文章1', '種文章2')
```

エスケープする必要がある種文章を LEN 指定なしで検索する場合

```
MCONCEPT RANK  
ファイル: eq('¥*はエスケープが必要です')
```

エスケープする必要がある種文章を LEN 指定で検索する場合



```
MCONCEPT RANK
ファイル: eq (LEN=23 '*はエスケープが必要です')
```

指定した条件に一致しない文章を検索する場合

```
MCONCEPT
NOT(ファイル: eq('種文章'))
```

見出しが種文章 1 に該当し、かつ本文が種文章 2 に該当する文書を検索する場合

```
MCONCEPT
AND(ファイル<章.見出し>:eq('種文章1'),ファイル<章.本文>:eq('種文章2'))
```

見出しが種文章 1 に該当するか、または本文が種文章 2 に該当しない文書を検索する場合

```
MCONCEPT
OR(ファイル<章.見出し>:eq('種文章1'),NOT(ファイル<章.本文>:eq('種文章2')))
```

種文章に該当するか、または検索タームが存在する文書を検索する場合

```
MCONCEPT
AND(ファイル: eq('種文章'),BASE(ファイル: eq('検索ターム')))
```

特徴タームを抽出しないで、指定した検索タームを検索する場合

```
MCONCEPT TERM
ファイル: eq('検索ターム')
```

## (6) 入力データ形式

サーチコマンド MCONCEPT は、入力データに概念検索の条件式を指定します。条件式の指定形式を次に示します。

検索条件式 | 複合条件式

検索条件式

検索条件式は、次の形式で記述します。

ファイル名 [ 構造名指定 [ [ 構造重み値 ] ] ] : eq ( 種文章指定 ) | BASE ( サーチコマンド MFIND の条件式 )

- ファイル名  
検索対象ファイルを指定します。
- 構造名指定  
特定の構造を指定して検索する場合は構造名を指定します。構造名は、次の形式で指定します。  
< 構造名 [ . 構造名 [ . 構造名 ] ... ] [ : 属性名 ] >
- 構造名  
検索する構造名を指定します。  
構造名は、テキストデータベースに登録された文書に存在する構造名だけを指定できます。このため、DTD に指定されていても文書に存在しない構造名を指定した場合は、エラーになります。
- 属性名  
構造に指定された属性を条件にして検索する場合、構造に対する属性名を指定し

#### 4. サーチコマンドの文法

ます。

- [ 構造重み値 ]  
検索条件式中の構造に重み（重要度）を付ける場合に指定します。複合条件式を使用して複数の検索条件式を記述する場合で、ランキング機能を使用するときに指定します。  
構造重み値は、1 ~ 100 の整数を [ ] で囲んで指定します。数値が大きいほど、重要度も上がります。省略した場合は1が仮定されます。
- 種文章指定  
検索対象となる種文章を指定します。種文章は次の形式で指定します。  
種文章〔重み指定〕[ < . 種文章〔重み指定〕 > ... ]
- 種文章  
検索したい種文章を指定します。次の形式で指定します。  
'種文章' | LEN= 数値 '種文章'
- '種文章'  
検索したい種文章を '( シングルクォーテーション ) で囲んで指定します。なお、次の文字を含む種文章を検索したい場合には、文字の前に必ず ¥ ( エスケープ文字 ) を指定してください。  
\* ? | ¥ ^ [ ] , ( ) '
- LEN= 数値 '種文章'  
検索したい種文章の長さを数値（バイト）で指定し、その直後に種文章を '( シングルクォーテーション ) で囲んで指定します。この指定では、エスケープ文字を指定する必要ありません。
- 重み指定  
ランキングオプションとターム指定オプションを指定して検索する場合で、それぞれの検索タームに重み（重要度）を付けたいときに指定します。重みは次の形式で指定します。  
[ 重み値 ]
- 重み値  
重みは 1 ~ 100 の数字で指定します。省略した場合には、1 が仮定されます。

#### 複合条件式

検索結果同士の論理演算を検索条件とします。複合条件式は、次の形式で指定します。

複合論理演算子（要素 [ ( , 要素 ) ... ]

- 複合論理演算子  
要素または要素間の検索条件を指定します。次の複合論理演算子を指定します。
- AND  
要素同士の論理積を検索結果とします。AND を指定する場合は、二つ以上の要素を指定してください。
- OR  
要素同士の論理和を検索結果とします。OR を指定する場合は、二つ以上の要素を

指定してください。

- NOT

要素の排除（補集合）を検索結果とします。指定した要素以外の文書を検索する場合に指定します。

要素は一つ指定してください。NOT を指定した場合は、ランキングオプションは無視されます。

- 要素

検索の条件にする検索結果集合、または条件式を指定します。要素は、次の形式で指定します。

検索条件式 | 複合条件式

なお、NOT 指定をした場合には、ランキングオプションを指定しても無視されず。

## (7) 実行結果の出力形式

MCONCEPT を指定した場合の実行結果の出力イメージを次に示します。

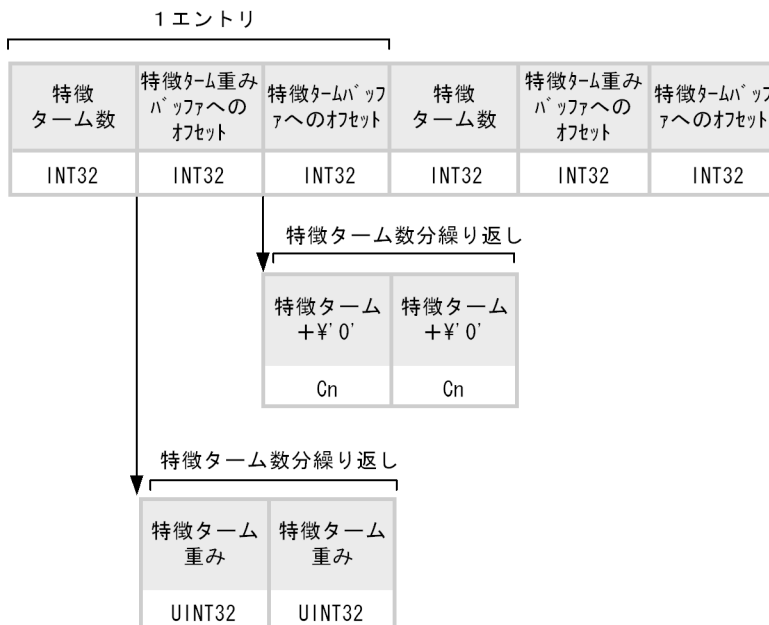
#### 4. サーチコマンドの文法

基点  
↓

種文章数分  
繰り返し

ベース検索 結果集合ID	検索結果 集合ID	ヒット件数	情報フラグ	表示可能 文書数	種文章数	A
INT32	INT32	INT32	UINT32	INT32	INT32	

A



(凡例)

Cn : nバイトの文字列型の配列

INT32 : 32ビット整数型

UINT32 : 32ビット符号なし整数型

注 入力データは1メガバイト以内で作成してください。

#### 構造体の定義

```
typedef struct TS2T_SEED_INFO {
    long    lTermnum;
    long    lScoreoff;
    long    lTermoff;
} TS2T_SEED_INFO;
typedef struct TS2T_SCMDOT_MCONCEPT {
    long    lBaseid;
    long    lNewid;
```

```
long    lHitcnt ;
unsigned long ulFlag ;
long    lDisplaycnt ;
long    lSeedcnt ;
struct TS2T_SEED_INFO seedinfo[1] ;
} TS2T_SCMDOT_MCONCEPT ;
```

### 4.3.3 M DATAMA KE ( 文書の追加 )

#### ( 1 ) 形式

M DATAMA KE ファイル名 [ ,REPLACE ]

#### ( 2 ) 機能

テキストデータベース ( 検索対象ファイル , および関連情報ファイル ) に文書を追加します。このとき , 構造なし文書用テキストデータベース , および構造文書用テキストデータベースのどちらにでも文書を追加できます。

#### ( 3 ) オプション

ファイル名

文書を追加する検索対象ファイル名 , または関連情報ファイル名を指定します。

REPLACE

すでに登録されている文書の内容を置き換えたいときに指定します。REPLACE を指定すると , 指定した文書 ID と同一の文書 ID の文書と置き換えます。REPLACE の指定がない場合 , 指定した文書 ID が登録されているとエラーになります。

#### ( 4 ) 注意事項

このコマンドを実行する前にサーチコマンド MSTART を実行して , 使用するテキストデータベースの検索セッションを開始しておいてください。

文書データ中に不正データがあった場合や , REPLACE 指定なしで既存文書 ID に登録しようとした場合は , TS2F\_cmd\_DATA 関数 , または TS2F\_rcvdat 関数の戻り値が -200001 になります。エラーの詳細については , 実行結果のステータスで確認してください。

0 バイトの文書は登録できません。

送信データ種別が「M DATAMA KE 用入力データ」でない場合は , TS2F\_sesopt() で送信データ種別を「M DATAMA KE 用入力データ」に設定してから実行してください。

テキストデータベース種別が XML の場合は , 文書データに全文検索用データを設定してください。全文検索用データの設定については「付録 D XML ファイルを全文検索用データファイルに変換する」を参照してください。

#### ( 5 ) 指定例

指定例を次に示します。

ファイル「Web」に文書データを追加する場合

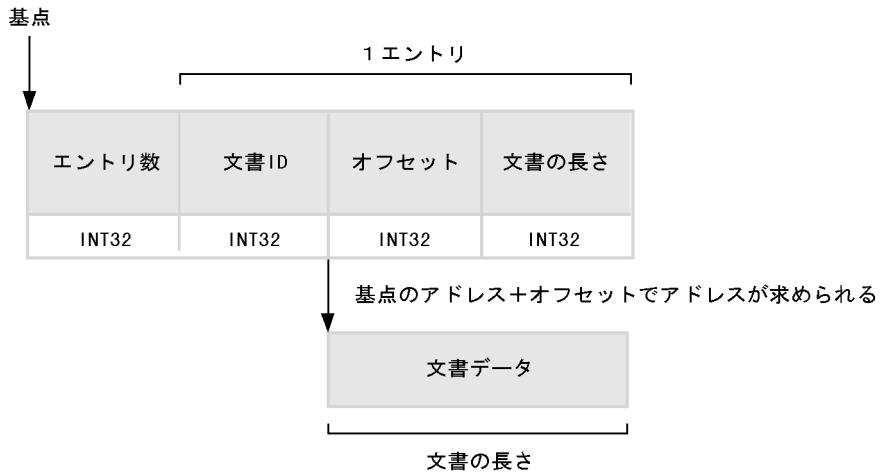
M DATAMA KE Web

ファイル「Web」に REPLACE を指定して文書データを追加する場合

MDATAMAKE Web, REPLACE

## (6) 入力データ形式

サーチコマンド MDATAMAKE の入力形式を次に示します。



(凡例)

INT32 : 32ビット整数型

注 入力データは1メガバイト以内で作成してください。

文書データは、エントリ部の次に続けて格納してください。

## 構造体の定義

```
typedef struct TS2T_SCMDIN_MDATAMAKE {
    long lEntry;
    struct {
        long lTextid;
        long lOffset;
        long lLength;
    } txtDir[1];
} TS2T_SCMDIN_MDATAMAKE;
```

## lEntry

エントリ数を指定します。1を指定してください。

## lTextid

文書IDを0～16,777,215の範囲で指定します。「-1」を指定すると、文書IDは自動的に採番されます。ただし、関連情報ファイルの場合は「-1」を指定できません。

## lOffset

文書データの位置を示す基点アドレスからの相対オフセット値を指定します。

#### 4. サーチコマンドの文法

lLength

文書データの長さを指定します。

txtidir

テキストディレクトリ（1 エントリ分のデータ）を示します。

### (7) 実行結果の出力形式

MDATAMAKE を指定した場合の実行結果の出力イメージを次に示します。

エントリ数	ステータス	エラー位置	エラー テキスト位置	文書ID
INT32	UINT32	INT32	INT32	INT32

(凡例)

INT32 : 32ビット 整数型

UINT32 : 32ビット符号なし整数型

#### 構造体の定義

```
typedef struct TS2T_SCMDOT_MDATAMAKE {
    long          lEntry;
    unsigned long ulStatus;
    long          lErrcnt;
    long          lPosition;
    struct {
        long          lTextid;
    } txtidir[1];
} TS2T_SCMDOT_MDATAMAKE;
```

lEntry

エントリ数（1）が設定されます。

ulStatus

実行結果のステータスが次に示す定数で設定されます。

- TS2D\_SCMD\_ASTAT\_NORMAL : 正常終了
- TS2D\_SCMD\_ASTAT\_DUPL : 指定した文書 ID はすでに登録されているが、REPLACE が指定されていない
- TS2D\_SCMD\_ASTAT\_DATER : 文書データ中に不正データが存在する

lErrcnt

エラーが発生した文書データのエン트리番号（1）が設定されます。



IPosition

エラーが発生したテキストの先頭からの位置が設定されます。

ITextid

文書 ID が設定されます。

### 4.3.4 MDATARM (文書の削除)

#### (1) 形式

MDATARM ファイル名, sdid (削除文書ID群)

#### (2) 機能

テキストデータベース (検索対象ファイル, および関連情報ファイル) の文書を削除します。このとき, 構造なし文書用テキストデータベース, および構造文書用テキストデータベースのどちらからでも文書を削除できます。

#### (3) オプション

ファイル名

文書を削除する検索対象ファイル名, または関連情報ファイル名を指定します。

削除文書 ID 群

削除する文章の文書 ID を, (半角コンマ) で区切って指定します。文書 ID を半角の - (ハイフン) で区切ると, その範囲の文書 ID が削除対象となります。

#### (4) 注意事項

このコマンドを実行する前に, サーチコマンド MSTART を実行して, 使用するテキストデータベースの検索セッションを開始しておいてください。

#### (5) 指定例

指定例を次に示します。

ファイル「Web」の文書 ID5 の文書を削除する場合

```
MDATARM Web, sdid(5)
```

ファイル「Web」の文書 ID 0 から 100 の文書を削除する場合

```
MDATARM Web, sdid(0-100)
```

### 4.3.5 MDBINF (テキストデータベース情報の出力)

#### (1) 形式

```
MDBINF { DB | DETAIL テキストデータベース名
          | STRUCT テキストデータベース名 }
```

#### (2) 機能

テキストデータベースに関する情報を出力します。

#### (3) オプション

DB

TS サーバーに登録されている、すべてのテキストデータベースの名称と数を取得する場合に指定します。

DETAIL テキストデータベース名

テキストデータベースの詳細情報を取得する場合に指定します。

STRUCT テキストデータベース名

テキストデータベースの構造情報を取得する場合に指定します。構造文書用テキストデータベースにだけ指定できます。

#### (4) 注意事項

オプションに STRUCT を指定して取得できる構造名は、テキストデータベースに登録された文書に存在する構造名だけを取得できます。このため、DTD に指定されていても、文書に存在しない構造名は取得できません。

構造名に半角の英小文字が含まれている場合、これらの文字は半角の英大文字に変換されます。

#### (5) 指定例

指定例を次に示します。

TS サーバーに登録されているテキストデータベースの名称と数を取得する場合

```
MDBINF DB
```

テキストデータベース「新聞記事」の詳細情報を取得する場合

```
MDBINF DETAIL 新聞記事
```

テキストデータベース「研究論文」の構造情報を取得する場合で、テキストデータベース「研究論文」は、構造文書用テキストデータベースであるとき

```
MDBINF STRUCT 研究論文
```

#### 4. サーチコマンドの文法

### (6) 実行結果の出力形式

#### (a) DB 句を指定した場合

DB 句を指定した場合の実行結果の出力イメージを次に示します。

データベースの定義数	データベース名	データベース名	...
INT32	C40	C40	

定義されている  
データベース分繰り返し

(凡例)

INT32 : 32ビット整数型

Cn : nバイトの文字型の配列

#### 構造体の定義

```
typedef struct TS2T_SCMDOT_MDBINF_DB {
    long lDbcnt;
    char caDbname[1][TS2DSZ_NAME];
} TS2T_SCMDOT_MDBINF_DB;
```

lDbcnt

TS サーバーに登録されているテキストデータベースの数が設定されます。

caDbname

テキストデータベース名が設定されます。lDbcnt での値分繰り返されます。

#### (b) DETAIL 句を指定した場合

DETAIL 句を指定した場合の実行結果の出力イメージを次に示します。

ファイル数分  
繰り返し

データ ベース名	辞書 フラグ	DB種別	予備1	登録された 最大文書ID	予備2	ファイル数	ファイル 情報 構造体
C40	C1	C1	C2	INT32	C44	INT32	

#### ファイル情報構造体

ファイル種別	検索対象ファイル構造体 または 関連情報ファイル構造体
INT32	

#### 検索対象ファイル構造体

検索対象 ファイル名	予 備
C40	C36

#### 関連情報ファイル構造体

関連情報 ファイル名	予 備
C40	C36

(凡例)

Cn : nバイトの文字型の配列

INT32 : 32ビット整数型

#### 構造体の定義

```
typedef struct TS2T_SCMDOT_MDBINF_FINDFILE {
    char caName[TS2DSZ_NAME];
    char caReserve[36];
} TS2T_SCMDOT_MDBINF_FINDFILE;
typedef struct TS2T_SCMDOT_MDBINF_INFFILE {
    char caName[TS2DSZ_NAME];
    char caReserve[36];
} TS2T_SCMDOT_MDBINF_INFFILE;
typedef struct TS2T_SCMDOT_MDBINF_FILE {
    long lFileType;
    union {
```

#### 4. サーチコマンドの文法

```
    TS2T_SCMDOT_MDBINF_FINDFILE tFind;
    TS2T_SCMDOT_MDBINF_INFFILE  tInf;
} unDetail;
} TS2T_SCMDOT_MDBINF_FILE;
typedef struct TS2T_SCMDOT_MDBINF_DETAIL {
    char          caDb[TS2DSZ_NAME];
    char          cDct;
    char          cDBType;
    char          caRsv[2];
    long          lMaxid;
    char          caRsv2[44];
    long          lFileEntry;
    TS2T_SCMDOT_MDBINF_FILE tFile[1];
} TS2T_SCMDOT_MDBINF_DETAIL;
```

##### TS2T\_SCMDOT\_MDBINF\_FINDFILE

検索対象ファイル構造体です。

##### caName

指定したテキストデータベースの検索対象ファイル名が設定されます。

##### caReserve

予備

##### TS2T\_SCMDOT\_MDBINF\_INFFILE

関連情報ファイル構造体です。

##### caName

指定したテキストデータベースの関連情報ファイル名が設定されます。

##### caReserve

予備

##### TS2T\_SCMDOT\_MDBINF\_FILE

ファイル情報構造体です。

##### lFileType

ファイルの種別が次に示す定数で設定されます。

- TS2D\_SCMD\_FILETYPE\_FIND：検索対象ファイル
- TS2D\_SCMD\_FILETYPE\_INF：関連情報ファイル

##### TS2T\_SCMDOT\_MDBINF\_DETAIL

DB 詳細情報を格納する構造体です。

##### caDb

テキストデータベース名が設定されます。

##### cDct

同義語辞書の設定の有無が設定されます。指定したテキストデータベースに同義語辞書が設定されている場合は、1 が出力されます。

##### cDBType

テキストデータベースの種別が次に示す定数で設定されます。

- TS2D\_SCMD\_DBTYPE\_STD : 構造を持たないテキストデータベース
- TS2D\_SCMD\_DBTYPE\_STRUCT : 構造を持つテキストデータベース
- TS2D\_SCMD\_DBTYPE\_XML : XML のテキストデータベース

caRsv

予備

IMaxid

登録された最大文書 ID が設定されます。

caRsv2

予備

lFileEntry

指定したテキストデータベースのファイル数が設定されます。

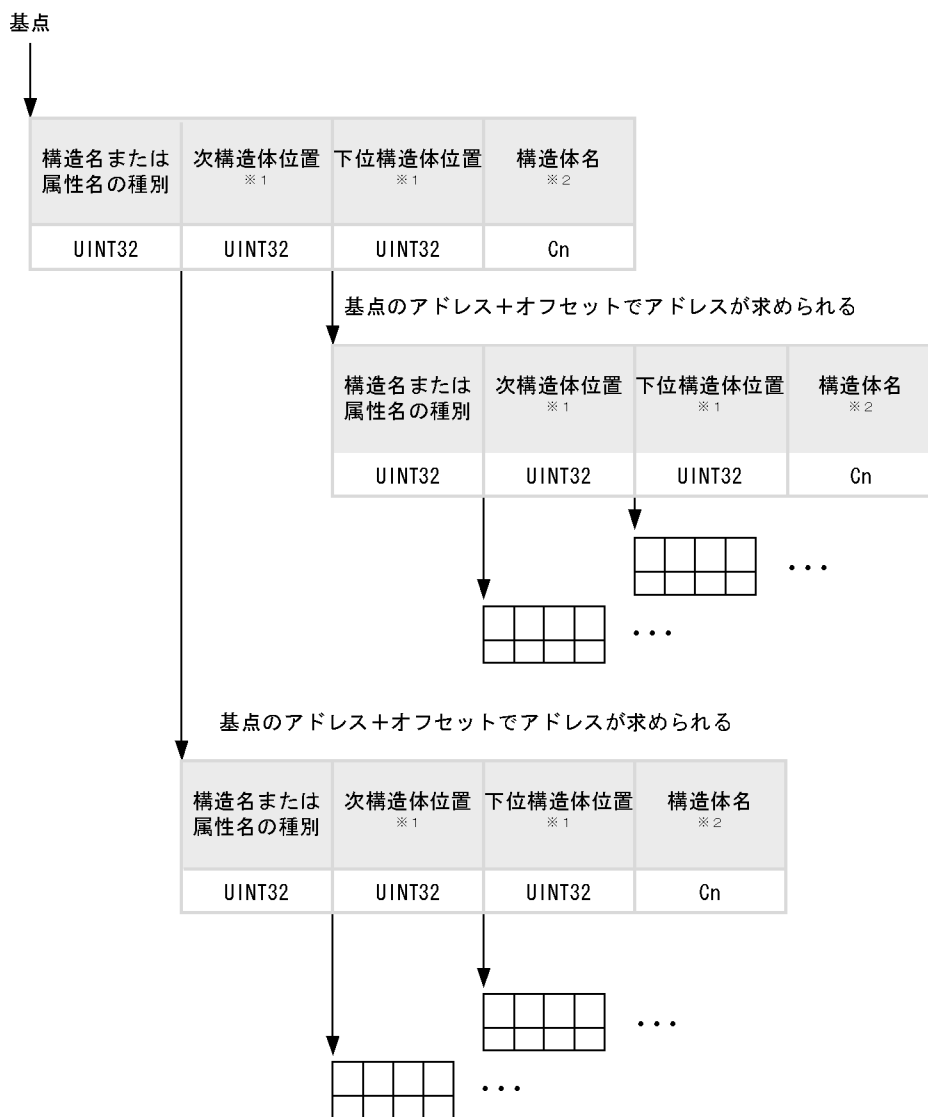
tFile

ファイル情報構造体です。lFileEntry での値分、構造体が存在します。

(c) STRUCT 句を指定した場合

STRUCT 句を指定した場合の実行結果の出力イメージを次に示します。

#### 4. サーチコマンドの文法



(凡例)

UINT32 : 32ビット符号なし整数型

Cn : nバイトの文字型の配列

注※1 続く構造体がない場合は0xffffffffが設定されます。

注※2 構造体名は((構造体名の長さ+3)/4\*4)になります。構造体名の長さには末尾の¥0も含まれます。

#### 構造体の定義

```
typedef struct TS2T_SCMDOT_MDBINF_STRUCT {
    unsigned long    ulType;
    unsigned long    ulNext;
    unsigned long    ulDown;
    char             caName[1];
} TS2T_SCMDOT_MDBINF_STRUCT;
```



**ulType**

caName に出力された文字列が構造名，または属性名のどちらであるかを判断する次の定数が設定されます。

- TS2D\_SCMD\_STTYPE\_STRUCT  
caName に出力された文字列は構造名です。
- TS2D\_SCMD\_STTYPE\_ATTR  
caName に出力された文字列は属性名です。

**ulNext**

構造体 TS2D\_SCMDOT\_MDBINF\_STRUCT がある場合，オフセットが設定されます。ない場合は，0xffffffff が設定されます。

**ulDown**

下位の構造体 TS2D\_SCMDOT\_MDBINF\_STRUCT のオフセットが設定されません。

**caName**

構造名，または属性名が文字列で設定されます。

### 4.3.6 MDISPLAY ( 検索結果の文書の出力 )

#### ( 1 ) 形式

MDISPLAY ファイル名, { 集合情報, outd(出力文書指定) | sdid(文書ID指定) }  
[ , tsize(文書範囲指定) ]

#### ( 2 ) 機能

検索した結果の文書を、テキストデータベースから取得します。

#### ( 3 ) オプション

##### ファイル名

取得する文書の検索対象ファイル名または関連情報ファイル名を指定します。

##### 集合情報

出力する文書の集合の形式を、次の形式で指定します。

{ set ( 検索結果集合 ID ) | RANK }

- set ( 検索結果集合 ID )

出力する文書が属する検索結果集合の ID を指定します。

- RANK

サーチコマンド MFIND でのランキング検索の結果を基に文書が出力されます。

したがって、RANK を指定する場合は、ランキングオプションを指定したサーチコマンド MFIND が実行されている必要があります。

RANK で指定した情報がソートされるかは、サーチコマンド MFIND のランキングオプションの指定によって決まります。

##### 出力文書指定

出力する文書を、次の形式で指定します。

- ALL

検索結果集合、またはランキング検索の結果に含まれるすべての文書が出力されます。

- 番号<sub>1</sub> - 番号<sub>2</sub>

検索結果集合、またはランキング検索の結果に含まれる番号<sub>1</sub>番目から番号<sub>2</sub>番目までの文書が出力されます。番号とは、検索結果集合またはランキング検索の結果で何番目の文書かを示します。

- 番号 [ { , 番号 } ... ]

番号に指定した文書が出力されます。番号とは、検索結果集合またはランキング検索の結果で何番目の文書かを示します。

##### 文書 ID 指定

出力する文書の文書 ID を、次の形式で指定します。

- 文書 ID [ { , 文書 ID } ... ]

指定された文書 ID に応じた文書が出力されます。

- 文書 ID<sub>1</sub> - 文書 ID<sub>2</sub>

文書 ID  $_1$  から文書 ID  $_2$  までの文書 ID に応じた文書が出力されます。

#### 文書範囲指定

出力する文書の範囲をバイト数で指定します。形式は次のとおりです。

- 位置 -  
指定した位置以降の文書が出力されます。
- [ - ] 位置  
指定した位置までの文書が出力されます。
- 位置 - 位置  
指定した位置の範囲での文書が出力されます。

#### (4) 注意事項

このコマンドを実行する前にサーチコマンド MSTART コマンドを実行して、使用するテキストデータベースの検索セッションを開始しておいてください。

OS が Windows の場合、サーチコマンド MDISPLAY で取得した文書を Windows の環境などで出力すると、表示が不正になることがあります。これは、サーチコマンド MDISPLAY で取得した文書の改行コード (0x0a) と Windows の環境での改行コード (0x0d + 0x0a) が異なるためです。必要に応じて、文書の改行コードを変換して出力してください。

#### (5) 指定例

指定例を次に示します。

検索結果集合 ID10 の集合に含まれる文書を「本文」ファイルから出力する場合  
MDISPLAY 本文 , set (10) , outd (ALL)

ランキング検索で得た結果に含まれる文書のうち、上位 50 文書を「見出し」ファイルから出力する場合  
MDISPLAY 見出し , RANK , outd (1-50)

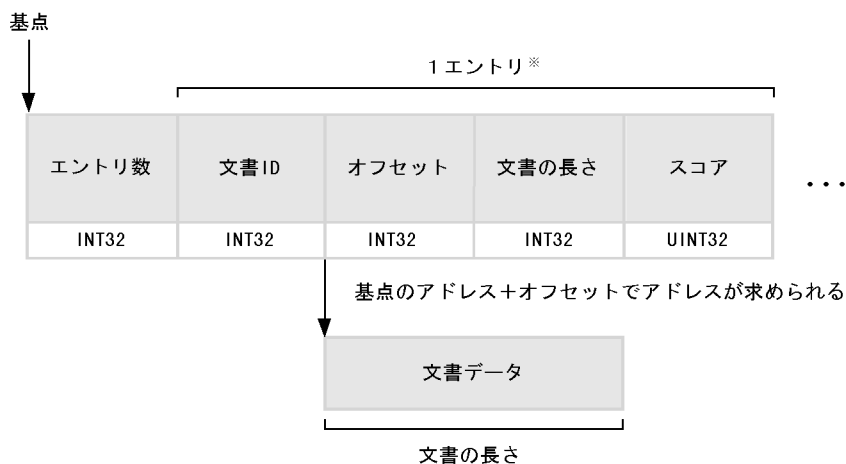
検索結果集合 ID10 の集合に含まれる文書のうち、先頭 20 バイトを「本文」ファイルから出力する場合  
MDISPLAY 本文 , set (10) , outd (ALL) , tsize (-20)

文書 ID15 の文書を「記事」ファイルから出力する場合  
MDISPLAY 記事 , sdid (15)

#### (6) 実行結果の出力形式

サーチコマンド MDISPLAY を指定した場合の実行結果の出力イメージを次に示します。

#### 4. サーチコマンドの文法



(凡例)

INT32 : 32ビット整数型

UINT32 : 32ビット符号なし整数型

注※ エントリ数分繰り返します。

#### 構造体の定義

```
typedef struct TS2T_SCMDOT_MDISPLAY {  
    long lTextcnt;  
    struct {  
        long lTextid;  
        long lOffset;  
        long lLength;  
        long lScore;  
    } txtidir[1];  
} TS2T_SCMDOT_MDISPLAY;
```

lTextcnt

取得した文書の件数が設定されます。

lTextid

取得した文書の文書 ID が設定されます。

lOffset

文書の格納位置を示す構造体の先頭からの相対オフセット値が設定されます。

lLength

文書の長さが設定されます。

lScore

ランキング検索での検索結果が存在する場合、文書に対するスコア値が設定されます。ランキング検索での結果が存在しない場合は、0 が設定されます。

txtidir

取得した文書に関する情報が設定されます。

### 4.3.7 MEND (テキストデータベースの使用終了宣言)

#### (1) 形式

MEND

#### (2) 機能

検索するテキストデータベースの使用を終了する宣言をして、検索セッションを終了します。

#### (3) 注意事項

すでに使用開始を宣言しているテキストデータベースに対して実行してください。

#### (4) 指定例

指定例を次に示します。

テキストデータベースの使用を終了する場合

MEND

## 4.3.8 MFIND (全文検索)

### (1) 形式

MFIND 条件式 [, ランキングオプション]  
[, 検索オプション [, 検索オプション] ...]

#### 条件式

検索条件式 | 複合条件式 | 特定構造検索条件式

- 検索条件式

ファイル名 [ 構造名指定 [[ 構造重み値 ]]] : 演算式

- 構造名指定

< 構造名 [. 構造名 [. 構造名] ...] [: 属性名] >

- 演算式

演算子 ( 項 [[, 項] ... ])

- 演算子

{ eq | fand | for | { c | C } { = | <= | >= } 距離 }

- 項

{ ' 検索ターム ' [[ 検索ターム重み値 ] ] | 演算式 }

- 複合条件式

複合論理演算子 ( 要素 [[, 要素] ... ])

- 複合論理演算子

{ AND | OR | NOT }

- 要素

{ SET ( 検索結果集合 ID ) | 検索条件式 | 複合条件式 | 特定構造検索条件式 }

- 特定構造検索条件式

SAND ( ファイル名 < 特定構造名 [[ . 特定構造名 ] ... ] > , 検索条件式 , 検索条件式 , [[ , 検索条件式 ] ... ] )

#### ランキングオプション

RANK [[ スコア算出方法 ] [ 論理演算指定時のスコア算出方法 ] [ ランキング詳細指定 ]]

- スコア算出方法

{ COUNTS }

- 論理演算指定時のスコア算出方法

{ BOOLEAN | TOTAL }

- ランキング詳細指定

{ SCORE | SORT [ 対象件数 [ FORCE ] ] }

#### 検索オプション

{ EXPANS [ 展開モード ] | [ BASE [ 検索結果集合 ID ] ] }

- 展開モード

#### 4. サーチコマンドの文法

[U][k][a][h][e]

### (2) 機能

指定した条件に従って、テキストデータベースを検索します。

### (3) オプション

#### 条件式

検索の条件を、次の形式で指定します。

検索条件式 | 複合条件式 | 特定構造検索条件式

#### 検索条件式

検索条件式は、次の形式で記述します。

ファイル名 [ 構造名指定 [ [ 構造重み値 ] ] ] : 演算式

- ファイル名

検索対象ファイルを指定します。

- 構造名指定

特定の構造を指定して検索する場合、構造名を指定します。構造名は、次の形式で指定します。

< 構造名 [ . 構造名 [ . 構造名 ] ... ] [ : 属性名 ] >

- 構造名

検索する構造名を指定します。

構造名は、テキストデータベースに登録された文書に存在する構造名だけを指定できます。このため、DTD に指定されていても文書に存在しない構造名を指定した場合は、エラーになります。

- 属性名

構造に指定された属性を条件にして検索する場合、構造に対する属性名を指定します。

- [ 構造重み値 ]

検索条件式中の構造に重み (重要度) を付ける場合に指定します。複合条件式を使用して複数の検索条件式を記述する場合で、ランキング機能を使用するときに指定します。

構造重み値は、1 ~ 100 の整数を [ ] で囲んで指定します。数値が大きいほど、重要度も上がります。省略した場合は1が仮定されます。

- 演算式

演算式は、次の形式で指定します。

演算子 ( 項 [ ( , 項 ) ... ] )

- 演算子

次の演算子を指定します。

- eq

項に指定した検索タームを検索します。項には文字列だけを指定できます。

- fand

項同士の論理積を検索結果とします。したがって、すべての項が存在または成立



することを条件にする場合に指定します。

and を指定する場合は、二つ以上の項を指定してください。さらに、項には検索タームまたは eq を除く演算式を指定してください。

- for

項同士の論理和を検索結果とするため、項の中で少なくとも一つが存在、または成立することを条件にする場合に指定します。

for を指定する場合は、二つ以上の項を指定してください。さらに、項には検索ターム、または eq を除く演算式を指定してください。

- {c|C}{= | <= | >=} 距離

項と項の間の文字列を条件にした検索（近傍条件検索）の場合に指定します。項は二つ指定してください。指定できる項は、検索タームまたは近傍条件を記述した演算式だけです。

なお、各演算子（c（または C）と =（または <= および >=）と距離）の間に空白などを記述しないでください。

演算子の意味を次の表に示します。

演算子	意味
c	項の順序を条件にしない場合に指定します。
C	項の順序を条件にする場合に指定します。 C を指定すると最初に指定した項に続いて、二つ目の項が現れることを条件とします。
=	項と項との間の文字列が、距離で指定する値と一致することを条件にする場合に指定します。
<=	項と項との間の文字列が、距離で指定する値以下であることを条件にする場合に指定します。
>=	項と項との間の文字列が、距離で指定する値以上であることを条件にする場合に指定します。
距離	項と項の間の文字数を 0 以上の整数で指定します。

- 項

検索の条件になる文字列（検索ターム）を指定します。検索タームの代わりに演算式を指定することもできます。

項は、次の形式で指定します。

'検索ターム' [[検索ターム重み値]] | 演算式

- 検索ターム

検索する文字列を '（アポストロフィ）で囲んで指定します。ただし、次に示す文字が検索タームに含まれる場合は、各文字の前に ¥（円記号）を指定してください

#### 4. サーチコマンドの文法

い。

\* ? | ¥ ^ [ ] , ( ) '

検索タームにはワイルドカード、および特殊文字を指定できます。ワイルドカード、および特殊文字は半角で指定してください。

指定できるワイルドカード、および特殊文字の意味を次の表に示します。

表 4-2 指定できるワイルドカード、および特殊文字

ワイルドカード、および特殊文字	意味
*	0文字以上の任意の文字列に相当します。
?	任意の1文字に相当します。
	構造の先頭または末尾に一致します。
¥	ワイルドカードまたは特殊文字の意味を消します。
^	直後にある1文字を検索対象から除外します。ただし、特殊文字をエスケープ文字(¥)と同時に指定した2文字は、1文字とみなして除外します。
[,]	除外する1文字を,(半角コンマ)で区切って2個以上指定する場合に使用します。

表 4-3 除外文字検索用の特殊文字の指定例

	指定例 意味
	ローマ^ 「ローマ?」から「ローマ字」の場合を除外して検索します
	^誤文書 「?文書」から「誤文書」の場合を除外して検索します
	データ^ベ^ー^ス 「データ??」から「データベース」の場合を除外して検索します
	テキスト^¥* 「テキスト?」から「テキスト*」の場合を除外して検索します
大阪^[府,市]	「大阪?」から「大阪市」と「大阪府」の場合を除外して検索します
文書^[の,を]管理	「文書?管理」から「文書の管理」と「文書を管理」を除外して検索します

**!** 注意事項

除外文字検索を使用する場合

- 除外対象文字だけを検索タームに指定すると、不正な検索条件としてエラーとなります。検索タームには、検索対象文字と除外対象文字を併せて指定してください。

ワイルドカード検索の場合も \* で区切られた個々の文字列に除外文字の指定をする場合は、検索対象文字を併せて指定する必要があります。

指定例を次に示します。

誤った例: ^ア, ^[ア, イ], ^ア\*, ^ア\*イ, ア\*^イ, ?^ア

正しい例: ^アイ, ^[ア, イ]ウ, イ^ア\*, ウ^ア\*イ, ア\*^イウ, ?^アイ

- 除外文字検索は同義語展開されません。
- 「^ 誤文書」を検索する場合、「? 文書」から「誤文書」を除外するので、構造または文書の先頭が「文書」で始まる文書はヒットしません。  
同じように「大阪 ^ 府」で検索した場合、構造または文書の終端が「大阪」の文書はヒットしません。

---

- [ 検索ターム重み値 ]

検索タームに重み（重要度）を付ける場合に指定します。fand, for および近傍条件検索で複数の検索タームを記述する場合で、ランキング機能を使用するときに指定できます。

検索ターム重み値は、1 ~ 100 の整数を [ ] で囲んで指定します。数値が大きいほど重要度も上がります。省略した場合は1が仮定されます。

## 複合条件式

検索結果同士の論理演算を検索条件とします。複合条件式は、次の形式で指定します。

複合論理演算子（要素 [[, 要素] ... ]）

- 複合論理演算子  
要素または要素間の検索条件を指定します。次の複合論理演算子を指定します。
- AND  
要素同士の論理積を検索結果とします。AND を指定する場合は、二つ以上の要素を指定してください。
- OR  
要素同士の論理和を検索結果とします。OR を指定する場合は、二つ以上の要素を指定してください。
- NOT  
要素の排他（補集合）を検索結果とするため、指定した要素以外の文書を検索する場合に指定します。  
要素は一つ指定してください。NOT を指定した場合は、ランキングオプションは無視されます。
- 要素  
検索の条件にする検索結果集合、または条件式を指定します。要素は次の形式で指定します。

#### 4. サーチコマンドの文法

- SET (検索結果集合 ID) | 検索条件式 | 複合条件式 | 特定構造検索条件式  
SET 句を指定した要素と、条件式を指定した要素を混在して記述できません。また、SET 句を指定した場合、ランキングオプション、および BASE 以外の検索オプションの指定は、無視されます。
- 検索結果集合 ID  
検索時に取得した検索結果集合 ID を指定します。

##### 特定構造検索条件式

複数の構造を対象として、指定した条件を満たす構造が、ある特定の構造に含まれるかどうかを検索する場合に指定します。特定構造検索条件式は、次の形式で指定します。

SAND (ファイル名 < 特定構造名 [( . 特定構造名 ) ... ] > , 検索条件式 , 検索条件式 , [( , 検索条件式 ) ... ])

- ファイル名  
検索対象ファイルを指定します。
- 特定構造名  
範囲を特定するための構造名を指定します。  
特定構造名は、テキストデータベースに登録された文書に存在する構造名だけを指定できます。このため、DTD に指定されていても文書に存在しない構造名を指定した場合は、エラーになります。

##### ランキングオプション

検索結果に重み (重要度) を付けて検索する場合に指定します。ランキングオプションを指定して検索すると、それまでのランキング検索の結果は消去されます。ランキングオプションは、次の形式で指定します。

RANK [( [スコア算出方法] [論理演算時スコア算出方法] [ランキング詳細指定] )]

- スコア算出方法  
ランキングのために与えるスコア (得点) の算出方法を指定します。スコア算出方法には COUNTS を指定します。
- COUNTS  
ヒットした検索タームの数、または成立した条件の数に基づいてスコアが付けられる方法です。

##### 論理演算指定時のスコア算出方法

論理演算時のスコア算出方法を指定します。論理演算時のスコア算出方法は次の形式で指定します。この指定を省略した場合、BOOLEAN が仮定されます。

BOOLEAN | TOTAL

- BOOLEAN  
BOOLEAN 方式でスコアの算出をします。検索条件が論理積のときは、演算対象 (検索タームまたは検索条件) の文書内でのそれぞれの出現数のうち、最小値に基づいてスコアが付けられる方式です。検索条件が論理和のときは、演算対象の文書内でのそれぞれの出現数のうち、最大値に基づいてスコアが付けられる方式です。

- TOTAL

スコア総和方式でスコアの算出をします。検索条件が論理積のときも論理和のときも、演算対象の文書内でのそれぞれの出現数の合計値に基づいてスコアが付けられる方式です。

#### ランキング詳細指定

ランキングの方法を指定します。ランキング詳細指定は、次の形式で指定します。指定を省略した場合は SCORE が仮定されます。

SCORE | SORT [対象件数 [FORCE]]

- SCORE

スコアを算出する場合に指定します。

- SORT [対象件数 [FORCE]]

ヒット件数が対象件数に指定した値以下の場合、算出されたスコアをキーとして降順にソートされます。対象件数を省略した場合は、1,000 が仮定されます。

FORCE を指定すると、ヒット件数が対象件数での値を超える場合でも降順にソートされ、上位から対象件数分の情報でランキングされます。

FORCE が指定されていない場合にヒット件数が対象件数を超えるときは、ランキングされません。

対象件数に大きな値を指定したり、FORCE を指定したときにヒット件数が多かったりすると、処理時間が長くなりますので注意してください。

#### 検索オプション

検索時の展開モード、またはベースを指定します。検索オプションを省略した場合は、サーチコマンド MOPT で指定された値が仮定されます。

検索オプションには EXPANS 句、または BASE 句を指定します。

- EXPANS [展開モード]

検索タームの展開モードを指定します。展開モードを省略すると、同義語・異表記展開検索されません。

展開モードは次の表に示す 5 種類を組み合わせで指定します。

展開モード	意味
U	検索タームを同義語展開する場合に指定します。ただし、同義語辞書が設定されていない場合は、エラーになります。
k	検索タームかたかな異表記展開する場合に指定します。
a	検索タームをアルファベット異表記展開する場合に指定します。
h	検索タームを半角アルファベット異表記展開する場合に指定します。
e	検索タームを全角・半角異表記展開する場合に指定します。

- BASE [検索結果集合 ID]

検索時のベースを指定します。

- 検索結果集合 ID

検索のベースにする検索結果集合の ID を指定します。ここで指定した検索結果集

#### 4. サーチコマンドの文法

合に含まれる文書が検索の対象になります。なお、0を指定すると、すべての文書が検索の対象になります。検索結果集合IDを省略すると0が仮定されます。

##### (4) 注意事項

このコマンドを実行する前に、サーチコマンド MSTART を実行して、使用するテキストデータベースの検索セッションを開始しておいてください。

##### (5) 指定例

検索条件式の記述例 1

- ファイル名「新聞記事」から「社会」および「政治」という単語がどちらも存在する文書を検索する場合

```
MFIND 新聞記事 :fand('社会','政治')
```

検索条件式の記述例 2

- 検索条件式の記述例 1 に「社会」に重みを付けてランキング検索する場合

```
MFIND 新聞記事 :fand('社会'[5],'政治'),RANK COUNTS SORT
```

複合条件式の記述例 1

- 構造名「見出し」に「消費税」が含まれる文書、および構造名「本文」に「5%」が含まれる文書を検索する場合

```
MFIND AND(新聞記事<文書.見出し>:eq('消費税'),新聞記事<文書.本文>:eq('5%'))
```

複合条件式の記述例 2

- 複合条件式の記述例 1 に、構造「本文」に重みを付けてランキング検索する場合

```
MFIND AND(新聞記事<文書.見出し>:eq('消費税'),  
新聞記事<文書.本文>[10]:eq('5%')),  
RANK COUNTS SORT 100 FORCE
```

特定構造検索条件式の記述例

- ファイル名「研究論文」から、章見出しに「イントラネット」を含んでいる場合で、章題に「ホームページ」を含む文書を検索するとき  
ただし「イントラネット」および「ホームページ」は、同じ章に存在することを条件とします。

```
MFIND SAND(研究論文<本文.章>,研究論文<本文.章.見出し>:eq('イントラネット'),研究論文<本文.章.章題>:eq('ホームページ'))
```

##### (6) 実行結果の出力形式

サーチコマンド MFIND (全文検索)を指定した場合の実行結果の出力イメージを次に示します。

ベースになる 検索結果集合 ID	検索結果集合 ID	ヒット件数	情報フラグ	ランキング情報 件数
INT32	INT32	INT32	UINT32	INT32

(凡例)

INT32 : 32ビット整数型

UINT32 : 32ビット符号なし整数型

### 構造体の定義

```
typedef struct TS2T_SCMDOT_MFIND {
    long          lBaseid;
    long          lNewid;
    long          lHitCnt;
    unsigned long ulFlag;
    long          lDisplaycnt;
} TS2T_SCMDOT_MFIND;
```

#### lBaseid

ベースの検索結果集合 ID が設定されます。

#### lNewid

検索結果集合 ID が設定されます。ヒットした文書がない場合は -1 が設定され、指定した検索条件に対する検索結果集合は作成されません。

#### lHitCnt

ヒットした文書の件数が設定されます。

#### ulFlag

次に示す定数が設定されています。状態を判定する場合は、値とマスク（ビットの論理積）を取り、0 以外の値になるかどうかを確認します。

マスク後、0 以外の値は次の状態を意味します。

- TS2D\_SCMD\_CREATE\_RANK : ランキング検索による検索結果が作成されました。

#### lDisplaycnt

TS サーバーが保持しているランキング情報の件数が設定されます。ランキング指定がない場合は、-1 が設定されます。

### 4.3.9 MFIND ( 範囲指定検索 )

#### ( 1 ) 形式

MFIND 関連情報ファイル名: rg ('範囲指定ターム') [, 検索オプション]

#### 検索オプション

BASE [ 検索結果集合 ID ]

#### ( 2 ) 機能

関連情報ファイルのテキストデータを検索タームとして、テキストデータベースを検索します。

#### ( 3 ) オプション

##### 関連情報ファイル名

検索対象となる関連情報ファイルを指定します。

##### 範囲指定ターム

検索範囲として次のどれか指定します。指定した検索タームも検索対象となりヒットします。

- 検索ターム 1, 検索ターム 2

検索ターム 1 と検索ターム 2 の範囲を検索します。二つの検索タームの間は、(コンマ) で区切ります。なお、検索ターム 1 と検索ターム 2 の文字列を文字コード順で前後逆に指定した場合、検索は無効になりヒットしません。

- 検索ターム 1,

指定した値以降を検索する場合は、1 個の検索タームのあとに、(コンマ) を指定します。

- , 検索ターム 1

指定した値以前を検索する場合は、1 個の検索タームの前に、(コンマ) を指定します。

範囲指定検索の検索タームには、ワイルドカード、および特殊文字は使用できません。また、次に示す文字が検索タームに含まれる場合は、各文字の前に ¥ を指定してください。

\* ? | ¥ ^ [ ] , ( ) '

##### 検索オプション

検索時のベースを指定します。指定を省略した場合は、サーチコマンド MOPT で指定された値が仮定されます。ただし、サーチコマンド MOPT の展開モード指定は無視されます。

BASE [ 検索結果集合 ID ]

検索時のベースを指定します。

- 検索結果集合 ID

検索のベースにする検索結果集合の ID または 0 を指定します。ここで指定した検



検索結果集合に含まれる文書 ID の関連情報登録データが検索の対象となります。0 を指定すると、すべての関連情報登録データが検索の対象になります。指定を省略した場合は、0 が仮定されます。

検索結果集合 ID には、全文検索結果、範囲指定検索のどちらでも指定できます。また、範囲指定結果集合の ID を、全文検索実行時に指定することもできます。BASE 指定を省略した場合は、ベースとなる検索結果集合は、サーチコマンド MOPT の SRCHMODE オプションに指定した値となります。

#### (4) 注意事項

このコマンドを実行する前に、サーチコマンド MSTART を使用して、使用するテキストデータベースの検索セッションを開始してください。

範囲指定検索の検索タームには、「付録 C 検索および登録に使用できる文字」の表 C-1 および表 C-2 に示す文字を使用してください。表 C-1 および表 C-2 に示す文字以外を使用した場合、期待した検索結果を取得できない場合があります。

範囲指定検索では、異表記展開および同義語辞書展開は実行できません。

範囲指定検索では、大小文字・全角半角文字統一機能を使用できません。

範囲指定検索の検索結果集合同士や、範囲指定検索と全文検索の検索結果集合を指定した複合論理演算（集合間論理演算）を実行できます。ただし、集合間論理演算では、検索対象ファイルに登録されていない文書 ID はヒットしません。このため、範囲指定検索で集合間論理演算を実行する場合は、関連情報ファイルに登録した文書 ID を検索対象ファイルにも登録しておく必要があります。

データが 1 件も登録されていない関連情報ファイルに対して範囲指定検索を実行すると、サーチコマンドはエラーステータス -200001（エラーマクロ名 TS2DERL\_ERR\_IN\_SERVER）となります。このときの TS サーバーのエラーコードは 2067 となります。

次の検索を実行した場合は、サーチコマンドはエラーステータス -200001（エラーマクロ名 TS2DERL\_ERR\_IN\_SERVER）となります。このときの TS サーバーのエラーコードは 2006 となります。

- 全文検索と範囲指定検索を同時に指定して実行した場合  
ただし、範囲指定検索時に全文検索結果集合 ID を BASE に指定すると、絞り込み検索が実行できます。同様に、全文検索時に範囲指定検索結果の集合 ID を BASE に指定した場合も絞り込み検索が実行できます。
- 複合条件式やランキング検索など、範囲指定検索条件式以外で範囲指定検索を実行した場合
- 範囲指定検索の検索タームにワイルドカードを指定して検索を実行した場合
- 検索対象ファイルに範囲指定検索を実行した場合
- 関連情報ファイルに対して全文検索を実行した場合
- 範囲指定検索の設定をしていない関連情報ファイルに対して範囲指定検索を実行した場合

#### 4. サーチコマンドの文法

##### (5) 指定例

関連情報ファイル名「日時データ」に、範囲指定検索用データとして次のフォーマットで文字列が登録されている場合を例に、指定例を示します。

フォーマット：YYYY/MM/DD hh:mm:ss

YYYY：西暦 4 けた

MM：月 2 けた (01 ~ 12)

DD：日 2 けた (01 ~ 31)

hh：時間 2 けた (00 ~ 23)

mm：分 2 けた (00 ~ 59)

ss：秒 2 けた (00 ~ 59)

- 西暦 2000 年の範囲指定検索の場合

MFIND 日時データ：rg ('2000/01/01 00:00:00,2000/12/31 23:59:59')

- 西暦 2001 年以降の検索の場合

MFIND 日時データ：rg ('2001/01/01 00:00:00,')

- 西暦 2000 年以前の検索の場合

MFIND 日時データ：rg ('',2000/12/31 23:59:59')

##### (6) 実行結果の出力形式

サーチコマンド MFIND (範囲指定検索) を指定した場合の実行結果の出力イメージを次に示します。

ベースになる 検索結果集合ID	検索結果集合ID	ヒット件数	情報フラグ	ランキング情報 件数
INT32	INT32	INT32	UINT32	INT32

(凡例)

INT32：32ビット整数型

UINT32：32ビット符号なし整数型

##### 構造体の定義

```
typedef struct TS2T_SCMDOT_MFIND {
    long          lBaseid;
    long          lNewid;
    long          lHitCnt;
    unsigned long ulFlag;
    long          lDisplaycnt;
} TS2T_SCMDOT_MFIND;
```

lBaseid

ベースの検索結果集合 ID が設定されます。

INewid

検索結果集合 ID が設定されます。ヒットした文書がない場合は -1 が設定され、指定した検索条件に対する検索結果集合は作成されません。

IHitCnt

ヒットした文書の件数が設定されます。

ulFlag

0 が設定されます。

IDisplaycnt

-1 が設定されます。

### 4.3.10 MINCREMENTAL (テキストデータベースのインクリメンタル (最適化))

#### (1) 形式

MINCREMENTAL [ term(文字列) [, force] ]

#### (2) 機能

指定した文字列をインクリメンタル (最適化) します。

#### (3) オプション

文字列

インクリメンタルを実行したい文字列を次の形式で実行します。

'文字列' [(,'文字列') ...]

文字列は、2文字以上3文字以下で指定します。文字列は100個まで指定できます。

なお、インクリメンタルを実行する文字列中に次の半角文字を含む場合は、各文字の前に¥を指定してください。

\* ? | ¥ ^ [ ] , ( ) '

- force

指定した文字列がインクリメンタルを実行する必要がないと判断できる場合でも、インクリメンタルを実行します。

#### (4) 注意事項

このコマンドを実行する前に、サーチコマンド MSTART を実行して、使用するテキストデータベースの検索セッションを開始しておいてください。

データベースに対して、データ登録 (サーチコマンド MDATAMAKE) が実行されている場合、このコマンドを実行するとエラーとなります。

サーチコマンドタイムアウト機能は、このコマンドに対して機能しません。

データベースのオープン時にメモリ常駐化定義ファイルを指定している場合、このコマンドを実行するとエラーとなります。

#### (5) 指定例

テキストデータベースをインクリメンタルする場合

```
MINCREMENTAL
```

### 4.3.11 MLOOK ( 検索タームの同義語・異表記展開の出力 )

#### (1) 形式

MLOOK [ EXPANS 展開モード, ] OUTMODE 出力モード, term( '検索ターム' [ , '検索ターム' ... ] )

展開モード

[ U | k | a | h | e ]

出力モード

[ ALL | COUNT ]

#### (2) 機能

指定した検索タームの同義語および異表記を展開して、結果を出力します。

#### (3) オプション

EXPANS [ 展開モード ]

検索タームの展開モードを指定します。EXPANS 句を省略すると、サーチコマンド MOPT で指定された値が仮定されます。展開モードを省略すると、同義語・異表記展開検索されません。

展開モード

展開モードは次の表に示す 5 種類を組み合わせで指定します。

展開モード	意味
U	検索タームを同義語展開する場合に指定します。ただし、同義語辞書が設定されていない場合は、エラーになります。
k	検索タームかたかな異表記展開する場合に指定します。
a	検索タームをアルファベット異表記展開する場合に指定します。
h	検索タームを半角アルファベット異表記展開する場合に指定します。
e	検索タームを全角・半角異表記展開する場合に指定します。

出力モード

検索タームの出力方法を指定します。出力モードは次のどちらかのモードを指定できます。

- ALL  
実行結果として、異表記展開した結果とその語数が出力されます。
- COUNT  
実行結果として、展開した語数だけが出力されます。

検索ターム

検索する文字列を ' ( アポストロフィ ) で囲んで指定します。ただし、次に示す文字が検索タームに含まれる場合は、各文字の前に ¥ を指定してください。

#### 4. サーチコマンドの文法

\* ? | ¥ ^ [ ] , ( ) '

検索タームには、ワイルドカードおよび特殊文字を指定できません。指定された場合は、次のように処理されます。

- \* または ? が指定された場合  
\* または ? を境にして、前後の二つの検索タームに分割されて処理されます。例えば、'ABC\*XYZ' と指定した場合は、'ABC' と 'XYZ' の二つの検索タームが指定されたものとして処理されます。
- | が指定された場合  
文字列の先頭または末尾に指定された場合は無視されます。それ以外の個所にある場合はエラーになります。

#### (4) 注意事項

このコマンドを実行する前に、サーチコマンド MSTART を実行して、使用するテキストデータベースの検索セッションを開始しておいてください。

出力モードに ALL を指定した場合、サーチコマンド MLOOK の実行結果は直接参照できません。サーチコマンド MLOOK の実行結果は、TS2F\_mlookcnv 関数で編集してください。

大小文字統一機能を指定しているデータベースに対してこのコマンドを実行した場合は、EXPANS ah を指定したものと仮定されます。

全角半角文字統一機能を指定しているデータベースに対してこのコマンドを実行した場合は、EXPANS Se を指定したものと仮定されます。

#### (5) 指定例

「コンピューター」の異表記展開の結果を取得する場合で、展開モードはサーチコマンド MOPT での設定値に従うとき

```
MLOOK OUTMODE ALL,term('コンピューター')
```

「コンピューター」および「パソコン」の同義語およびかたかな異表記展開の結果を取得する場合

```
MLOOK EXPANS Uk,OUTMODE ALL,  
term('コンピューター','パソコン')
```

「パソコン」の同義語展開した語数を取得する場合

```
MLOOK EXPANS U,OUTMODE COUNT,  
term('パソコン')
```

#### (6) 実行結果の出力形式

出力モードに COUNT を指定した場合の実行結果の出力イメージを次に示します。

同義語展開数
--------

INT32
-------

(凡例)

INT32 : 32ビット整数型

#### 構造体の定義

```
typedef struct TS2T_SCMDOT_MLOOK_COUNT {  
    long lTermcnt;  
} TS2T_SCMDOT_MLOOK_COUNT;
```

- lTermcnt

異表記展開された単語の数が設定されます。

#### ! 注意事項

出力モードに ALL を指定した場合の実行結果の出力イメージは、TS2F\_mlookenv 関数を参照してください。

---

### 4.3.12 MOPT ( 検索オプションの指定 )

#### ( 1 ) 形式

MOPT 検索オプション { [ , 検索オプション ] ... }

##### 検索オプション

{ EXPANS [ 展開モード ] | { SRCHMODE [ サーチモード [ 検索結果集合 ID ] ] }

- 展開モード  
{ U | k | a | h | e }
- サーチモード  
{ U | H }

#### ( 2 ) 機能

検索オプションを指定します。検索オプションには、次の 2 種類を指定できます。

- 展開モード  
同義語・異表記展開した結果も検索条件にする場合に指定します。
- サーチモード  
検索時のベース、およびベースの設定方法 ( サーチモード ) を指定します。

#### ( 3 ) オプション

##### 検索オプション

EXPANS 句、または SRCHMODE 句を指定します。

- EXPANS [ 展開モード ]  
検索タームの展開モードを指定します。展開モードを省略すると、同義語・異表記展開検索されません。
- 展開モード  
展開モードは次の表に示す 5 種類を組み合わせで指定します。

展開モード	意味
U	検索タームを同義語展開する場合に指定します。ただし、同義語辞書が設定されていない場合は、エラーになります。
k	検索タームかたかな異表記展開する場合に指定します。
a	検索タームをアルファベット異表記展開する場合に指定します。
h	検索タームを半角アルファベット異表記展開する場合に指定します。
e	検索タームを全角・半角異表記展開する場合に指定します。

SRCHMODE [ サーチモード [ 検索結果集合 ID ] ]

検索時のベースを指定します。SRCHMODE 句を省略すると、U 0 が仮定されます。

##### サーチモード

次のどちらかのモードを指定します。



- U  
ユニバースサーチで検索します。ユニバースサーチとは、ベースにする検索結果集合を固定して検索する方法です。
- H  
ハイアラキサーチで検索します。ハイアラキサーチとは、検索結果から検索を続け、検索結果を絞り込む方法です。

#### 検索結果集合 ID

検索のベースにする検索結果集合の ID を指定します。ここで指定した検索結果集合に含まれる文書が検索の対象になります。なお、0 を指定すると、すべての文書が検索の対象になります。検索結果集合 ID を省略すると 0 が仮定されます。

#### (4) 注意事項

このコマンドを実行する前に、サーチコマンド MSTART を実行して、使用するテキストデータベースの検索セッションを開始しておいてください。

#### (5) 指定例

検索オプションの展開モードに、かたかな異表記展開および全角・半角異表記展開を指定する場合

```
MOPT EXPANS ke
```

検索結果集合 ID が 10 の集合をベースにしてハイアラキサーチで検索し、更に同義語展開を指定する場合

```
MOPT SRCHMODE H 10,EXPANS U
```

### 4.3.13 MOSTAT ( 検索オプションの設定値の取得 )

#### ( 1 ) 形式

MOSTAT

#### ( 2 ) 機能

サーチコマンド MOPT で設定された検索オプションの設定値を取得します。

#### ( 3 ) 注意事項

このコマンドを実行する前に、サーチコマンド MSTART を実行して、使用するテキストデータベースの検索セッションを開始しておいてください。

#### ( 4 ) 指定例

サーチコマンド MOPT で設定した検索オプションの設定値を出力する場合

MOSTAT

#### ( 5 ) 実行結果の出力形式

サーチコマンド MOSTAT の実行結果の出力イメージを次に示します。

使用するデータベース名	ベースになる検索結果集合ID	異表記展開モード	サーチモード
C40	INT32	INT32	UINT32

( 凡例 )

Cn : nバイトの文字型の配列  
 INT32 : 32ビット整数型  
 UINT32 : 32ビット符号なし整数型

#### 構造体の定義

```
typedef struct TS2T_SCMDOT_MOSTAT {
    char          caDbname [TS2DSZ_NAME];
    long         lBaseid;
    unsigned long ulExpans;
    unsigned long ulSrchmode;
} TS2T_SCMDOT_MOSTAT;
```

#### caDbname

使用しているテキストデータベースの名称が設定されます。

#### lBaseid

ベースの検索結果集合 ID が設定されます。

#### ulExpans

次の表に示す定数が設定されています。状態を判定する場合は、値とマスク

(ビットの論理積)を取り, 0以外の値になるかどうかを確認します。

定数	意味
TS2D_SCMD_EXP_N_DOUGI	同義語展開
TS2D_SCMD_EXP_N_KATA	かたかな異表記展開
TS2D_SCMD_EXP_N_ALPHA	アルファベット異表記展開
TS2D_SCMD_EXP_N_HALPHA	半角アルファベット異表記展開
TS2D_SCMD_EXP_N_EXCHG	全角・半角異表記展開

#### ulSrchmode

サーチモードが次に示す定数で設定されます。

- TS2D\_SCMD\_SMODE\_H : ハイアラーキモード
- TS2D\_SCMD\_SMODE\_U : ユニバースモード

### 4.3.14 MSORT ( 検索結果のソート )

#### ( 1 ) 形式

MSORT SET (検索結果集合ID), 関連情報ファイル名 [, {ASC | DESC} ]  
[ , LIMIT 上限件数 ]

#### ( 2 ) 機能

検索結果を、指定した関連情報ファイルに登録されているテキストデータでソートします。

ソートキーとなるテキストデータは、関連情報ファイルに登録されている各データの先頭から 32 文字までです。「付録 C 検索および登録に使用できる文字」の表 C-1 に示す文字以外の文字がある場合は、その文字の直前までがソートキーとして使用されます。データが 32 文字未満の場合は、登録されているだけのデータでソートします。また、2 バイト文字コードも 1 バイト文字コードもそれぞれ 1 文字として扱われます。

このコマンドの実行結果は、サーチコマンド MFIND の実行結果と同様に、検索結果集合に格納されます。このため、検索結果集合 ID はサーチコマンド MSORT、および MFIND の実行結果を通して、一意となります。

このコマンドの実行結果である検索結果集合 ID を指定して、サーチコマンド MDISPLAY を実行すると、ソートされた順番で文書を取得できます。

#### ( 3 ) オプション

検索結果集合 ID

ソートする検索結果集合 ID を指定します。

関連情報ファイル名

ソート対象とする関連情報ファイル名を指定します。

ASC | DESC

ソート方法を指定します。指定を省略した場合は、ASC が仮定されます。

- ASC

昇順のソート結果を取得する場合に指定します。ソートキーとなるテキストデータが同じ場合は、文書 ID の昇順にソートされます。関連情報ファイルにソートキーとなるデータが登録されていない文書 ID は、ソート結果の後方に文書 ID の昇順に並びます。

- DESC

降順のソート結果を取得する場合に指定します。キーとなるテキストデータが同じ場合は、文書 ID の降順にソートされます。関連情報ファイルにソートキーとなるデータが登録されていない文書 ID は、ソート結果の後方に文書 ID の降順に並びます。

LIMIT 上限件数

ソート対象となる検索結果集合 ID の結果に含まれるヒット件数（ソート対象件数）の上限値を、0 以上の整数で指定します。ソート対象件数が上限値を超えている場合はソートを実行しないで、サーチコマンドはエラーステータス -200001（エラーマクロ名 TS2DEL\_ERR\_IN\_SERVER）となります。このときの TS サーバーのエラーコードは 2082 となります。0 を指定した場合は、ソート対象件数に関係なく、ソートを実行します。  
指定を省略した場合は、10,000 が仮定されます。

#### （4）注意事項

このコマンドを実行する前にサーチコマンド MSTART を実行して、使用するテキストデータベースの検索セッションを開始しておいてください。

LIMIT に 0 や大きな値を設定したときにソート対象件数が多かった場合、処理時間が長くなります。

#### （5）指定例

検索結果集合 ID3 の検索結果を「登録日時」関連情報ファイルで昇順にソートする場合

```
MSORT SET(3),登録日時,ASC
```

または

```
MSORT SET(3),登録日時
```

検索結果集合 ID5 の検索結果を「登録日時」関連情報ファイルで降順にソートする場合

```
MSORT SET(5),登録日時,DESC
```

検索結果集合 ID10 のソート対象件数が 1,500 件以内の場合で「登録日時」関連情報ファイルで昇順にソートするとき

```
MSORT SET(10),登録日時,ASC,LIMIT 1500
```

または

```
MSORT SET(10),登録日時,LIMIT 1500
```

検索結果集合 ID15 のソート対象件数が何件の場合でも「登録日時」関連情報ファイルで降順にソートする場合

```
MSORT SET(15),登録日時,DESC,LIMIT 0
```

#### （6）実行結果の出力形式

サーチコマンド MSORT の実行結果の出力イメージを次に示します。

#### 4. サーチコマンドの文法

ベース検索 結果集合 ID	検索結果 集合 ID	対象検索	情報フラグ	有効文書数
INT32	INT32	INT32	INT32	INT32

(凡例)

INT32 : 32ビット整数型

#### 構造体の定義

```
typedef struct TS2T_SCMDOT_MSORT {  
    long      lBaseid;  
    long      lNewid;  
    long      lBaseCnt;  
    unsigned long ulFlag;  
    long      lSortCnt;  
} TS2T_SCMDOT_MSORT;
```

lBaseid

ベースの検索結果集合 ID が設定されます。

lNewid

ソート結果の検索結果集合 ID が設定されます。

lBaseCnt

ソート対象になった件数が設定されます。

ulFlag

0 が設定されます。

lSortCnt

ソート対象の文書 ID の中で、関連情報ファイルにデータが登録されていた有効文書数です。

#### (7) 実行例

ソートの実行例を次に示します。

## ●ソート前のデータ

文書ID	キー文字列
311	2001/10/09
205	2001/10/09 13:24
121	(未登録)
95	平成10年02月
30	(未登録)
45	2001/10/09
289	平成09年12月
163	2001/10/13 09:30

IBaseCntの設定値 : 8  
ISortCntの設定値 : 6



## ●昇順のソート結果

文書ID	キー文字列
45	2001/10/09
311	2001/10/09
205	2001/10/09 13:24
163	2001/10/13 09:30
289	平成09年12月
95	平成10年02月
30	(未登録)
121	(未登録)

## ●降順のソート結果

文書ID	キー文字列
95	平成10年02月
289	平成09年12月
163	2001/10/13 09:30
205	2001/10/09 13:24
311	2001/10/09
45	2001/10/09
121	(未登録)
30	(未登録)

### 4.3.15 MSTART (テキストデータベースの使用開始宣言)

#### (1) 形式

MSTART テキストデータベース名

#### (2) 機能

検索するテキストデータベースの使用を開始する宣言をして、検索セッションを開始します。

#### (3) オプション

テキストデータベース名

使用するテキストデータベース名を指定します。

#### (4) 注意事項

使用するテキストデータベースは、あらかじめオープンしておく必要があります。テキストデータベースはサーバーでオープンします。テキストデータベースのオープンについては、マニュアル「Bibliotheca2 TextSearch Version 2 システム管理者ガイド」を参照してください。

サーチコマンド MDBINF 以外のサーチコマンドを使用する場合は、先にサーチコマンド MSTART を実行してください。

すでに検索セッションが開始しているテキストデータベースにサーチコマンド MSTART を実行するとエラーになります。

#### (5) 指定例

「議事録」という名称のテキストデータベースを使用する場合

MSTART 議事録



### 4.3.16 MSWEEP ( 検索結果集合の削除 )

#### (1) 形式

MSWEEP [ 検索結果集合ID [, 検索結果集合ID] ... ]

#### (2) 機能

検索結果集合を削除します。

#### (3) オプション

##### 検索結果集合 ID

削除する検索結果の検索結果集合 ID を指定します。検索結果集合 ID を省略した場合は、すべての検索結果が削除されます。

0 または現在の検索セッションにある最大の検索結果集合 ID を超える値が指定された場合は、エラーになります。

#### (4) 注意事項

このコマンドを実行する前に、サーチコマンド MSTART を実行して、使用するテキストデータベースの検索セッションを開始しておいてください。

すべての検索結果集合を削除した場合またはベースになる検索結果集合を削除した場合は、ベースの検索結果集合 ID に 0 が設定されます。

#### (5) 指定例

##### 検索結果をすべて削除する場合

MSWEEP

##### 検索結果集合 ID 5 および 6 の検索結果を削除する場合

MSWEEP 5,6



# 付録

---

付録 A Library の環境設定

---

付録 B ユーザープログラムの作成例

---

付録 C 検索および登録に使用できる文字

---

付録 D XML ファイルを全文検索用データファイルに変換する

---

付録 E 関数のエラーステータス

---

付録 F TS サーバーのエラーコード

---

付録 G 用語解説

---

---

## 付録 A Library の環境設定

ここでは、Library のインストール後の環境設定について、OS ごとに説明します。

### 付録 A.1 OS が Windows の場合の環境設定

#### (1) ポート番号の設定

インストールの終了後、services ファイルに対して Library が使用するポート番号の定義を追加する必要があります。

追加する行を次に示します。なお、行末には改行コードを入力してください。

services ファイルのパス

OS のインストール先ディレクトリ ¥system32¥drivers¥etc¥services

追加する行

bib2ts 20025/tcp

TS サーバーにアクセスするアプリケーションプログラムをインストールしたマシンの services ファイルにも、同一のポート番号の設定が必要です。ポート番号の定義が無い場合は、上記のポート番号を使用して動作します。

#### (2) インストールパス

Library を新規にインストールした場合のデフォルトインストールパスを次に示します。なお、Runtime Library の場合インストールパスは固定です。

Development Kit または Server Library の場合

- 32bit OS の場合  
OS ドライブ : ¥Program Files¥HITACHI¥Bib2TSDK
- 64bit OS の場合  
OS ドライブ : ¥Program Files (x86)¥HITACHI¥Bib2TSDK

Runtime Library の場合

- 32bit OS の場合  
OS ドライブ : ¥Program Files¥HITACHI¥Bib2TSLIB
- 64bit OS の場合  
OS ドライブ : ¥Program Files (x86)¥HITACHI¥Bib2TSLIB

### 付録 A.2 OS が UNIX の場合の環境設定

#### (1) 通信環境の設定

インストールの終了後、通信環境を設定してください。設定内容を次に示します。

### hosts ファイルの設定

/etc 下の hosts ファイルに、接続するサーバーマシンの IP アドレスを記述します。

### services ファイルの設定

/etc 下の services ファイルに、Server Library が使用するポート番号の定義を追加します。追加する行を次に示します。なお、行末には、改行コードを入力してください。

```
bib2ts    20025/tcp
```

TS サーバーにアクセスするアプリケーションプログラムをインストールしたマシンの services ファイルにも、同一のポート番号を設定してください。ポート番号の定義が無い場合は、上記のポート番号を使用して動作します。

## (2) インストールパス

Library をインストールした場合のインストールパスを次に示します。なお、インストールパスは固定です。

```
/opt/Bib2TS/TS2/library
```

## 付録 A.3 文字コード種別の環境設定

接続するテキストデータベースの文字コード種別を環境変数 TSLANG で設定します。文字コード種別の設定をしない場合は、シフト JIS が設定されたものとしします。設定する値について次の表に示します。

表 A-1 TSLANG の指定値

クライアント側の文字コード種別	テキストデータベースの文字コード種別	TSLANG の値
シフト JIS コード	sjis	sjis
UTF-8 コード	utf-8 または utf-8_ucs4 (JISX0221)	utf-8
	utf-8-ms または utf-8-ms_ucs4 (MS-Unicode)	utf-8-ms

## 付録 A.4 通信ソケットの環境設定

TS サーバーとの通信で使用するソケットの使用方法を、環境変数 TS2SOCKET で設定できます。

通信ソケットの使用方法は、次の二つから選択します。

- アクション単位ソケット  
セッション確立、セッション切断、サーチコマンド実行のアクション単位ごとにソケットの生成、およびクローズを繰り返します。

- セッション単位ソケット

セッション確立でソケット生成します。セッション切断までソケットをクローズしません。

アクション単位ソケットは通信ソケットを必要な場合だけ確保するので、アクションがないときは通信ソケットを消費しません。しかし短時間にアクションを繰り返した場合、OS の設定によっては OS 側の通信ソケットの解放が間に合わず、通信ソケットの生成に失敗する場合があります。

セッション単位ソケットは、セッション確立からセッション切断まで一つの通信ソケットを維持します。このため、通信ソケットを消費したままになります。通信ソケット生成の失敗がほとんどありません。

アクションが頻繁に発生する場合は、セッション単位ソケットを設定した運用を推奨します。

アクション単位ソケットとセッション単位ソケットの設定を次の表に示します。

表 A-2 通信ソケットの設定

環境変数名	設定する値	説明
TS2SOCKET	EACH	アクション単位ソケットを使用する場合に指定します。デフォルトの値です。
	ONCE	セッション単位ソケットを使用する場合に指定します。

- 値を指定しない場合、または不正な値を指定した場合は、デフォルトの設定で動作します。
- 通信ソケットの設定は、UAP が動作するマシンで行ってください。
- 通信ソケットの設定は、TS2F\_con 関数を発行した時点で有効になります。セッション確立後に値を変更した場合でも、セッションを切断するまでセッション確立時に設定した値が有効となります。

## 付録 B ユーザープログラムの作成例

図 B-1 ~ 図 B-8 に Development Kit または Server Library を使った UAP の作成例を示します。この UAP は、次の手順を実行するプログラムです。

1. 入力した検索タームを検索する
2. 該当した文書を一覧表示する
3. 一覧の中から、文書の内容をテキスト表示する

図 B-1 ユーザープログラムの作成例 ( 1/8 )

```

/*****
/* Bibliotheca2/TS クライアントサンプルプログラム          */
/*                                                         */
/* サーバ名      = myhost                                */
/* DB名         = newspaper                              */
/* ファイル名    = article                              */
/*                                                         */
/*****
#include <stdio.h>
#include <stdlib.h>
#include <ctype.h>
#include "TS2lib.h"

#define SVname    "myhost"                                /* 接続サーバ名称          */
#define DBname    "newspaper"                            /* 検索DB名称              */
#define FILEname  "article"                              /* 検索ファイル名称        */

#define LIST_SIZE 60                                    /* 結果一覧で表示する文書のバイト数 */

long main();                                           /* メイン関数              */
long listmode(long, long, long);                       /* 文書一覧選択関数        */
void listout( long, long, TS2T_SCMDOT_MDISPLAY * );    /* 文書一覧表示関数        */
long textout(long, long, long);                        /* 文書表示関数            */
void errorprint();                                     /* エラー情報出力関数      */

typedef struct {                                       /* MDISPLAYエントリ構造体 */
    long    ITextid;                                   /* 文書ID                  */
    long    IOffset;                                  /* 文書データの対オフセット */
    long    ILength;                                  /* 文書データの長さ        */
    long    IScore;                                   /* スコア                  */
} mdisplay_ent;

```

図 B-2 ユーザープログラムの作成例 (2/8)

```

/*****
/* メイン関数
/*****
long main()
{
    long IWsess = -1;                /* 通信セッション識別子          */
    long IWcommand;                 /* コマンド識別子受け取り用      */
    long IWrsiz;                    /* 実行結果サイズ受け取り用      */
    char caWcommand[128];           /* サーチコマンド文字列バッファ  */
    char caWterm[BUFSIZ];          /* 検索タームバッファ            */
    TS2T_SCMDOT_MFIND tWFindResult; /* 検索実行結果                  */
    long IWrt;                      /* 戻り値受け取り用              */

    /* 通信セッション開始 */
    IWsess = TS2F_con( SVname );
    if( IWsess<0 ){
        fprintf( stderr, "通信セッションが開始できません。エラー番号は%dです。¥n", IWsess );
        errorprint();
        goto end;
    }
    /* 検索セッション開始 */
    sprintf( caWcommand, "MSTART %s", DBname );
    IWrt = TS2F_cmd( IWsess, &IWcommand, caWcommand, NULL, 0, &IWrsiz );
    if( IWrt<0 ){
        fprintf( stderr, "検索セッションが開始できません。エラー番号は%dです。¥n", IWrt );
        errorprint();
        goto end;
    }
    /* 検索オプション設定 */
    IWrt = TS2F_cmd( IWsess, &IWcommand, "MOPT EXPANS ka", NULL, 0, &IWrsiz );
    if( IWrt<0 ){
        fprintf( stderr, "MOPT失敗です。エラー番号は%dです。¥n", IWrt );
        errorprint();
        goto end;
    }
}

```



図 B-3 ユーザープログラムの作成例 (3/8)

```

/* 検索 */
while( 1 ){
    printf( "検索タームを入力してください : " );
    gets( caWterm );
    /* MFIND実行 */
    sprintf( caWcommand, "MFIND %s:eq('%s')", FILEname, caWterm );
    lWrtn = TS2F_cmd( lWsess, &lWcommand, caWcommand, &tWFindResult,
        sizeof(TS2F_SCMDOT_MFIND), &lWsize );
    /* 検索失敗 */
    if( lWrtn<0 ){
        fprintf( stderr, "MFIND失敗です。エラー番号は%dです。¥n", lWrtn );
        errorprint();
        /* 0件ヒット */
    } else if( tWFindResult.lHitCnt<=0 ){
        printf( "見つかりませんでした。¥n" );
        /* ヒットしたら、文書一覧選択 */
    } else {
        lWrtn = listmode( lWsess, tWFindResult.lHitCnt, tWFindResult.lNewid );
        if( lWrtn!=1 ){
            break;
        }
    }
}
/* 検索セッション終了 */
lWrtn = TS2F_cmd( lWsess, &lWcommand, "MEND", NULL, 0, &lWsize );
if( lWrtn<0 ){
    fprintf( stderr, "検索セッションが終了できません。エラー番号は%dです。¥n", lWrtn );
    errorprint();
    goto end;
}
end:
/* 通信セッションを開始していたら終了する*/
if( lWsess>=0 ){
    lWrtn = TS2F_discon( lWsess );
    if( lWrtn<0 ){
        fprintf( stderr, "通信セッションが終了できません。エラー番号は%dです。¥n", lWrtn );
        errorprint();
    }
}
printf( "お疲れ様でした。¥n" );
return 0;
}

```

図 B-4 ユーザープログラムの作成例 (4/8)

```

/*****
/* 文書一覧選択関数 */
/* 戻り値 : 0:検索終了 1:検索続行 -1:異常終了 */
/*****
long listmode( long lXsess, long lXhitcnt, long lXsetid )
{
    TS2T_SCMDOT_MDISPLAY *ptWmdisplay;          /* MDISPLAY結果バッファ */
    long lWtxtnum;                               /* テキスト数 */
    long lWcommand;                             /* コマンド識別子受け取り用 */
    long lWrsiz;                                /* 実行結果サイズ受け取り用 */
    char caWcommand[128];                       /* サーチコマンド文字列バッファ */
    char caWinbuf[BUFSIZ];                     /* 入力バッファ */
    long lWTextid;                              /* テキストID */
    long lWrtn;                                  /* 戻り値受け取り用 */

    /* MDISPLAY用バッファ(1カ^ハ^イ)取得 */
    ptWmdisplay = (TS2T_SCMDOT_MDISPLAY*)malloc(TS2DSZ_1MB);
    if( ptWmdisplay==NULL ){
        fprintf( stderr, "メモリ不足です。¥n" );
        return -1;
    }
    /* MDISPLAY実行(全文書の先頭LIST_SIZE+1[byte]取得) */
    sprintf( caWcommand, "MDISPLAY %s, set (%d), outd (ALL), tsize (%d)", FILEName, lXsetid,
LIST_SIZE+1 );
    lWrtn = TS2F_cmd( lXsess, &lWcommand, caWcommand, ptWmdisplay, TS2DSZ_1MB, &lWrsiz );
    if( lWrtn<0 ){
        fprintf( stderr, "MDISPLAY失敗です。エラー番号は%dです。¥n", lWrtn );
        errorprint();
        free( ptWmdisplay );
        return -1;
    }
    /* 取得できたテキスト数を設定 */
    lWtxtnum = ptWmdisplay->lTextcnt;

```

図 B-5 ユーザープログラムの作成例 (5/8)

```

/* 文書一覧表示 */
listout( lXsess, lXhitcnt, ptWmdisplay );
/* 処理選択ループ */
while( 1 ){
    printf( "一覧:h 検索ターム入力:t 終了:q 文書表示:文書番号 ? " );
    gets( caWinbuf );
    /* h なら処理選択ループを抜け、一覧出力へ戻る */
    if( caWinbuf[0]=='h' ){
        listout( lXsess, lXhitcnt, ptWmdisplay );
        continue;
    } /* t なら検索ターム入力へ return 1 する */
    } else if( caWinbuf[0]=='t' ){
        free( ptWmdisplay );
        return 1;
    } /* q なら終了処理へ return 0 する */
    } else if( caWinbuf[0]=='q' ){
        free( ptWmdisplay );
        return 0;
    } /* 数字で始まるなら文書番号として処理する */
    } else if( caWinbuf[0]>'0' && caWinbuf[0]<='9' ){
        lWTextid = atoi( caWinbuf );
        /* 文書番号範囲チェック */
        if( lWTextid<=0 || lWTextid>lWtxtnum ){
            printf( "文書番号が不正です。1~%dの範囲で指定してください。¥n", lWtxtnum );
            continue;
        }
    }
} /* 文書表示 */
lWrtn = textout( lXsess, lXsetid, lWTextid );
if( lWrtn<0 ){
    free( ptWmdisplay );
    return 0;
}
}
}
}

```

図 B-6 ユーザープログラムの作成例 (6/8)

```

/*****
/* 文書一覧表示関数 */
/* 戻り値 : 0:検索終了 1:検索続行 -1:異常終了 */
/*****
void listout( long lXsess, long lXhitcnt, TS2T_SCMDOT_MDISPLAY *ptXmdisplay )
{
    mdisplay_ent *ptWent;                /* MDISPLAYエントリ */
    long lWtxtnum;                       /* テキスト数 */
    long lWtxtcnt;                       /* テキストカウンタ */
    unsigned char *pcWtxt;              /* テキストデータポインタ */

    /* MDISPLAYできた件数, MDISPLAYエントリ取得 */
    lWtxtnum = ptXmdisplay->lTextcnt;
    ptWent = (mdisplay_ent*)(ptXmdisplay->ttxtdir);
    /* 文書一覧表示 */
    printf( "ヒット件数 : %d件\n", lXhitcnt );
    printf( "文書番号 : 文 書\n" );
    for( lWtxtcnt=0; lWtxtcnt<lWtxtnum; lWtxtcnt++){
        printf( "%8d : ", lWtxtcnt+1 );
        pcWtxt = (unsigned char*)ptXmdisplay + ptWent[lWtxtcnt].lOffset;
        printf( "%. *s\n", ptWent[lWtxtcnt].lLength, pcWtxt );
    }
    return;
}

```

図B-7 ユーザープログラムの作成例(7/8)

```

/*****
/* 文書表示関数
/* 戻り値 : 0:正常終了 -1:異常終了
/*****
long textout( long IXsess, long IXsetid, long IXTextid )
{
    TS2T_SCMDOT_MDISPLAY *ptWmdisplay;          /* MDISPLAY結果バッファ          */
    mdisplay_ent *ptWent;                       /* MDISPLAYエントリ              */
    long lWcommand;                             /* コマンド識別子受け取り用      */
    char *pcWtxt;                               /* テキストデータポインタ        */
    char caWcommand[128];                      /* サーチコマンド文字列バッファ  */
    long lWrsize;                               /* 実行結果サイズ受け取り用      */
    long lWrtn;                                 /* 戻り値受け取り用              */

    /* MDISPLAY用バッファ(1kバイト)取得 */
    ptWmdisplay = (TS2T_SCMDOT_MDISPLAY*) malloc(TS2DSZ_1MB);
    if( ptWmdisplay==NULL ){
        fprintf( stderr, "メモリ不足です。¥n" );
        return -1;
    }
    /* MDISPLAY実行(指定文書の全データ取得) */
    sprintf( caWcommand, "MDISPLAY %s, set(%d), outd(%d)", FILEName, IXsetid, IXTextid );
    lWrtn = TS2F_cmd( IXsess, &lWcommand, caWcommand, ptWmdisplay, TS2DSZ_1MB, &lWrsize );
    if( lWrtn<0 ){
        fprintf( stderr, "MDISPLAY失敗です。エラー番号は%dです。¥n", lWrtn );
        errorprint();
        free( ptWmdisplay );
        return -1;
    }
    /* MDISPLAY結果情報取得 */
    ptWent = (mdisplay_ent*)(ptWmdisplay->tctxdir);
    pcWtxt = (char*)ptWmdisplay + ptWent->lOffset;
    /* 文書表示 */
    printf( "¥n----- 文書番号 %d -----¥n", IXTextid );
    fwrite( pcWtxt, 1, ptWent->lLength, stdout );
    printf( "¥n" );

    free( ptWmdisplay );
    return 0;
}

```

図 B-8 ユーザープログラムの作成例 ( 8/8 )

```

/*****
/* 詳細エラー表示関数
/*****
void errorprint()
{
    char caWmessage[256];
    long lWerror;
    long lWrtn;

    caWmessage[0] = 0x0;
    lWrtn = TS2F_geterror( caWmessage, 256, &lWerror );

    if( lWrtn==TS2DERL_SERVER ){
        printf( "サーバでエラーが発生しました。¥n" );
        printf( "エラーコード %d¥n", lWerror );
        printf( "エラー情報 %s¥n", caWmessage );
    } else if( lWrtn==TS2DERL_CLIENT ){
        printf( "クライアントでエラーが発生しました。¥n" );
        printf( "エラーコード %d¥n", lWerror );
        printf( "エラー情報 %s¥n", caWmessage );
    } else {
        printf( "エラー詳細情報はあります。¥n" );
    }
    return;
}

```

## 付録 C 検索および登録に使用できる文字

### (1) シフト JIS コードで使用できる文字

テキストデータベースの文字コード種別が sjis の場合、Bibliotheca2 TS で登録、および検索に使用できる文字を次の表に示します。

表 C-1 登録するテキストデータに使用できる文字

文字種	使用できる文字コード (16 進数)	
	1 バイト目	2 バイト目
1 バイト文字コード	0x09 0x20 ~ 0x7E 0xA1 ~ 0xDF	-
2 バイト文字コード	0x81 ~ 0x9F 0xE0 ~ 0xFC	0x40 ~ 0x7E 0x80 ~ 0xFC

(凡例) - : 該当しません。

注 上記以外の文字が登録文書中に含まれていても登録エラーにはなりません。ただし、上記以外の文字は検索できません。

### (2) UTF-8 コードで使用できる文字

テキストデータベースの文字コード種別が utf-8, utf-8-ms, utf-8\_ucs4, utf-8-ms\_ucs4 のどれかの場合、Bibliotheca2 TS で登録、および検索に使用できる文字を次の表に示します。

表 C-2 UTF-8 コードで使用できる文字

文字種	使用できる文字コード (16 進数)
1 バイト文字コード	0x09 0x20 ~ 0x7E
2 バイト文字コード	UCS コード U+0080 ~ U+07FF を UTF-8 にエンコードした文字コード
3 バイト文字コード	UCS コード U+0800 ~ U+FFFF を UTF-8 にエンコードした文字コード
4 バイト文字コード	UCS コード U+010000 ~ U+10FFFF を UTF-8 にエンコードした文字コード

注 1 上記以外の文字が登録文書中に含まれていても登録エラーにはなりません。ただし、上記以外の文字は検索できません。

注 2 文字コード種別が utf-8, または utf-8-ms のデータベースに 4 バイト文字コードの文字が含まれる文書を登録した場合、エラーにはなりませんが 4 バイト文字コードの文字は検索できません。4 バイト文字コードを検索する場合は、データベースの文字コード種別に utf-8\_ucs4, または utf-8-ms\_ucs4 を指定してデータベースを作成してください。

注 3 登録する文書が XML 文書の場合、表 C-2 の範囲の文字を encoding 属性に従い変換したデー

タが使用できる文字です。

### (3) 使用できない文字に関する注意

Bibliotheca2 TS では、文字列中に使用できる文字以外（無効文字）が含まれている場合は、検索タームを指定してもヒットしません。ただし、無効文字が登録文書中に含まれている場合でも、登録は正常に処理されます。このため、文字列中に無効文字が含まれている場合は注意が必要です。無効文字が不可視文字として含まれているときの検索を例に説明します。

(例) 連続した文字列中に、無効文字が不可視文字として含まれているときの検索

次の文字列を登録したとします。

- 表示されている文字列：日立
- 実際のデータ：日<無効文字>立

この場合、「日」と「立」の間に<無効文字>が含まれているため、1文字目は「日」、<無効文字>は検索対象ではないので無視され、2文字目は「立」として登録されます。つまり、連続した文字列「日立」ではなく、「日」と「立」の独立した文字として登録されます。

このとき、検索タームに「日立」を指定して検索を実行しても、登録データが「日」と「立」の独立した文字のためにヒットしません。また、検索タームに<無効文字>指定しても、<無効文字>は登録されていないためヒットしません。

このため、「日<無効文字>立」をヒットさせるには、検索タームを次のように指定してください。

- 「日?立」
- 「日\*立」
- 「日」と「立」の近傍条件

### (4) UTF-8 コードの JISX0221 と MS-Unicode の違い

UTF-8 コードで運用するデータベース作成時の文字コード種別には、JISX0221 準拠 (utf-8 または utf-8\_ucs4) と、MS-Unicode 準拠 (utf-8-ms または utf-8-ms\_ucs4) のどちらかを選択する必要があります。ただし、UTF-8 コードの JISX0221 と MS-Unicode では、一部の文字の割り当てが違うため Bibliotheca2 TS の動作に影響を与える場合があります。ここでは JISX0221 と MS-Unicode の Bibliotheca2 TS への影響について、それぞれ説明します。影響する内容を確認してから、登録データに使用されている文字コードに合わせたデータベースの文字コード種別を選択してください。

(a) 全半角文字統一機能および全角・半角異表記展開への影響

Bibliotheca2 TS には、同じ文字を全角文字と半角文字で区別なく検索できるようにするため、データベース作成時に指定する全半角文字統一機能、および検索実行時に指定する全角・半角異表記展開が存在します。しかし、JISX0221 と MS-Unicode では全角文字と半角文字の相互変換に対応する文字コードの一部に違いがあります。このため、違いがある文字を使用して登録・検索を行う場合、JISX0221 または MS-Unicode に合致



しない文字コードはヒットしません。

違いがある文字ごとに対応する UTF-8 の文字コードを、次の表に示します。

表 C-3 JISX0221 と MS-Unicode の文字別全半角対応 UTF-8 文字コード

文字	仕様	半角文字コード	全角文字コード
- (ハイフン)	JISX0221	0x2D ( U+002D )	0xE28892 ( U+2212 )
	MS-Unicode		0xEFBC8D ( U+FF0D )
¥ (円記号)	JISX0221	0xC2A5 ( U+00A5 )	0xEFBA5 ( U+FFE5 )
	MS-Unicode	0x5C ( U+005C )	
~ (チルダ)	JISX0221	0xE280BE ( U+203E )	0xEFBA3 ( U+FE3 )
	MS-Unicode	0x7E ( U+007E )	

注 括弧内は、UCS-2 の文字コードを示します。

表 C-3 に示す文字を全半角文字統一機能、または全角・半角異表記展開を使用して登録、検索を実行する場合にヒットするかを次の表に示します。

表 C-4 登録文字コードと検索ターム文字コードの組み合わせによるヒット有無

対象文字	登録文字コード	データベースの文字コード種別	検索タームの文字コード	ヒット
- (ハイフン)	0x2D	utf-8	0x2D	
			0xE28892	
			0xEFBC8D	×
		utf-8-ms	0x2D	
			0xE28892	×
			0xEFBC8D	
	0xE28892	utf-8	0x2D	
			0xE28892	
			0xEFBC8D	×
		utf-8-ms	0x2D	×
			0xE28892	
			0xEFBC8D	×
0xEFBC8D	utf-8	0x2D	×	
		0xE28892	×	
	utf-8-ms	0xEFBC8D		
		0x2D		

付録 C 検索および登録に使用できる文字

対象文字	登録文字コード	データベースの文字コード種別	検索タームの文字コード	ヒット
		utf-8-ms_ucs4	0xE28892	×
			0xEFBC8D	
¥ (円記号)	0xC2A5	utf-8	0xC2A5	
		utf-8_ucs4	0x5C	×
			0xEFBA5	
		utf-8-ms	0xC2A5	
		utf-8-ms_ucs4	0x5C	×
			0xEFBA5	×
	0x5C	utf-8	0xC2A5	×
		utf-8_ucs4	0x5C	
			0xEFBA5	×
		utf-8-ms	0xC2A5	×
		utf-8-ms_ucs4	0x5C	
			0xEFBA5	
0xEFBA5	utf-8	0xC2A5		
	utf-8_ucs4	0x5C	×	
		0xEFBA5		
	utf-8-ms	0xC2A5	×	
	utf-8-ms_ucs4	0x5C		
		0xEFBA5		
~ (チルダ)	0xE280BE	utf-8	0xE280BE	
		utf-8_ucs4	0x7E	×
			0xEFBA3	
		utf-8-ms	0xE280BE	
		utf-8-ms_ucs4	0x7E	×
			0xEFBA3	×
	0x7E	utf-8	0xE280BE	×
		utf-8_ucs4	0x7E	
			0xEFBA3	×
		utf-8-ms	0xE280BE	×
		utf-8-ms_ucs4	0x7E	
			0xEFBA3	
0xEFBA3	utf-8	0xE280BE		
	utf-8_ucs4	0x7E	×	

対象文字	登録文字コード	データベースの文字コード種別	検索タームの文字コード	ヒット
			0xEFBF3	
		utf-8-ms	0xE280BE	x
		utf-8-ms_ucs4	0x7E	
			0xEFBF3	

(凡例)

○ : ヒットします。

x : ヒットしません。

(b) エスケープ文字への影響

Bibliotheca2 TS では辞書ソースファイル中、または検索条件式中の特殊文字を通常文字として扱う場合、特殊文字の前にエスケープ文字として ¥ (円記号) を追加します。ただし、辞書ソースファイル中には全角と半角の ¥ を、検索条件式中には半角の ¥ をエスケープ文字として追加します。

¥ は JISX0221 と MS-Unicode で半角文字コードに違いがあるため、データベースの文字コード種別に合わせて ¥ の文字コードを指定する必要があります。文字コード種別が一致しない場合は、エスケープ文字ではなく通常文字として処理されます。

¥ をエスケープ文字、または通常文字のどちらで取り扱うかを次の表に示します。

表 C-5 ¥ (円記号) の取り扱い

対象	データベース文字コード種別	指定文字コード	取り扱い
辞書ソースファイル中	utf-8	0xC2A5	エスケープ文字
		0x5C	通常文字
		0xEFBF3	エスケープ文字
	utf-8-ms	0xC2A5	通常文字
		0x5C	エスケープ文字
		0xEFBF3	エスケープ文字
検索条件式中	utf-8	0xC2A5	エスケープ文字
		0x5C	通常文字
	utf-8-ms	0xC2A5	通常文字
		0x5C	エスケープ文字

## 付録 D XML ファイルを全文検索用データファイルに変換する

XML ファイルを全文検索用データファイルに変換する場合は、TS2F\_well\_formed\_xmlcpl 関数、または TS2XMLcncv コマンドを使用します。TS2F\_well\_formed\_xmlcpl 関数の詳細については「3. 関数の文法」を参照してください。ここでは、TS2XMLcncv コマンドについて説明します。

### (1) 形式

```
TS2XMLcncv -h 全文検索用データ変換定義ファイル名
            -e エラー出力ファイル名
            -m 文字コード種別
            [-r]
```

### (2) 機能

XML ファイルを全文検索用データファイルに変換します。

### (3) 引数

- h 全文検索用データ変換定義ファイル名  
全文検索用データ変換定義ファイル名を 200 バイト以内のフルパスで指定します。全文検索用データ変換定義ファイルの詳細については、次項の「(4) XML 文書の制限」を参照してください。
- e エラー出力ファイル名  
エラー情報を出力するファイル名を 200 バイト以内のフルパスで指定します。既存のファイル名を指定した場合は、既存のファイル内容の最後にエラー情報を出力します。
- m 文字コード種別  
出力する全文検索用データの文字コードを指定します。指定できる文字コード種別は utf-8、utf-8-ms です。  
文字コード種別は、登録するデータベースの文字コード種別に応じて指定値を決定します。指定する文字コード種別を次に示します。

データベースの文字コード種別	全文検索用データの文字コード種別
utf-8	utf-8
utf-8_ucs4	
utf-8-ms	utf-8-ms
utf-8-ms_ucs4	

-r

同じ名称のファイルがすでに存在している場合は、新しいファイルに置き換えます。

#### (4) XML 文書の制限

TS2XMLcnv コマンドで変換できる XML 文書は、Cosminexus XML Processor が処理できる範囲です。Cosminexus XML Processor については、マニュアル「Cosminexus XML Processor ユーザーズガイド」を参照してください。

XML 文書を変換する場合は、次の点に注意してください。

##### XML 文書のサイズ

入力できる 1 文書のサイズは、最大 3MB です。ただし、変換した全文検索用ファイルのサイズが 1MB を超える場合、MDATAMAKE コマンドを使用した全文検索用データファイルの登録はできません。1MB を超える全文検索用データファイルを登録する場合は、TS サーバーのユティリティコマンド TS2DTcpl を使用して登録してください。

ユティリティコマンド TS2DTcpl の詳細については、マニュアル「Bibliotheca2 TextSearch Version 2 システム管理者ガイド」を参照してください。

##### XML 文書の構造

複数の XML 文書を一つのテキストデータベースに登録する場合は、最上位タグ名称を同じにしてください。下位の構造は同じにする必要はありません。なお、テキストデータベースの最上位タグ名称は、テキストデータベースを作成したときに指定した構造検証用ファイルの最上位タグ名称です。

##### タグ名称

タグ名称に半角英小文字が含まれている場合は、半角英大文字として扱われます。大文字小文字の区別はされません。

##### タグのネスト数

タグのネスト数（深さ）は 100 を上限とします。ただし、属性が指定できるネスト数（深さ）は 99 が上限です。

##### XML 文書のエンコーディング属性

XML 文書の文字コードは、XML パーサや XSLT トランスフォーマの入出力 XML 文書に記述する XML 宣言の encoding 属性（encoding="XXX" の XXX の部分）に指定します。Bibliotheca2 TS で指定できる encoding 属性は次のとおりです。

- UTF-8
- UTF-16
- UTF-16BE
- UTF-16LE
- US-ASCII
- ISO-10646-UCS-2

##### 補足説明

encoding 属性が UTF-8 以外の XML 文書に登録した場合でも、クライアントで

指定する検索条件式の文字コードや、クライアントが取得する文書データの文字コードは UTF-8 となります。

#### XML 文書の仕様

XML 文書は次の仕様で解析を行いません。

- 外部一般エンティティおよび、外部パラメタエンティティの取り込みを行いませんので、指定した URI が参照できるようにしてください。
- XML1.0 の名前（プレフィックス付き）と属性（xmlns 宣言も属性として扱います）を有効にします。
- 妥当性制約の検証は行いません。

なお、Bibliotheca2 TS では HTTP の外部参照はサポートしていません。XML 文書に HTTP の外部参照が記載されている場合、Cosminexus XML Processor が自動的に接続します。ネットワーク環境により接続ができない場合は、接続のタイムアウトまで無応答になります。このため、XML 文書に HTTP の外部参照が記載されている場合は、記載を削除することをお勧めします。

#### (5) 全文検索用データ変換定義ファイルの記述形式

全文検索用データ変換定義ファイルは次の形式で記述します。

##### 記述形式

```
0 入力する XML ファイル名 0, 0 出力する全文検索用データファイル名
0<改行>
[ 0 入力する XML ファイル名 0, 0 出力する全文検索用データファイル名
0<改行 ] ...
```

##### 説明

- 入力する XML ファイル名  
登録する XML 文書ファイル名をフルパスで指定します。ディレクトリおよびファイル名は、空白、タブ、半角コンマ以外の ASCII コードで指定してください。
- ,(半角コンマ)  
入力する XML ファイル名と出力する全文検索用データファイル名の区切り文字として、(半角コンマ)を指定します。
- 出力する全文検索用データファイル名  
出力する全文検索用データファイル名をフルパスで指定します。ディレクトリおよびファイル名は、空白、タブ、半角コンマ以外の ASCII コードで指定してください。

##### 記述上の注意事項

- \*(半角アスタリスク)で始まる行はコメント行とみなします。
- 行に空白やタブ、または改行しかない場合は無視します。
- 一つの全文検索用データ変換定義ファイル（入力する XML ファイル名と出力する全文検索用データファイル名）は、一行に記述してください。

- 行末には改行を指定してください。

#### 全文検索用データ変換定義ファイルの構文

全文検索用データ変換定義ファイルの記述例を次の図に示します。

図 D-1 全文検索用データ変換定義ファイルの記述例 (OS が Linux の場合)

```
*コメント
/MyDocument/001.xml,/MyDocument/001.esisb<改行>
/MyDocument/245.xml△,△/MyDocument/245.esisb<改行>
△/MyDocument/100.xml,/MyDocument/100.esisb<改行>
△/MyDocument/101.xml,/MyDocument/101.esisb<改行>
△/MyDocument/102.xml,/MyDocument/102.esisb<改行>
△/MyDocument/103.xml,/MyDocument/103.esisb<改行>
```

(凡例)

<改行>:改行コード

### (6) コマンドの格納先

コマンドの格納先は使用する OS により異なります。OS 別の格納先を次に示します。

OS が Windows の場合

インストールディレクトリ ¥bin

OS が UNIX の場合

/opt/Bib2TS/TS2/library/bin

### (7) メモリ使用量

TS2XMLCnv の最大メモリ使用量を指定する場合は、環境変数 TS2XMLCNV\_MAXMEM を使用してください。環境変数の指定をしない場合は、Java のデフォルト値で動作します。Java のデフォルト値の詳細については、マニュアル「Cosminexus システム設計ガイド」を参照してください。なお、TS2XMLCnv は Client VM で動作します。

メモリ所要量の指定値の計算式を次に示します。

$$\text{TS2XMLCNV\_MAXMEM} = \max(\text{XML ファイルの最大サイズ [MB]} \times 100, 64)$$

上記の値を設定しても、「KIBS9000-E MEMORY SHORTAGE」のメッセージが表示された場合は、TS2XMLCNV\_MAXMEM に次の値を指定してください。

$$\text{TS2XMLCNV\_MAXMEM} = \text{使用可能な仮想メモリ最大値 [MB]}$$

TS2XMLCNV\_MAXMEM の値が不正、または大きすぎる場合は、「KIBS」が付かないエラーメッセージを表示します。

## (8) 注意事項

- TS2XMLcnv コマンドを実行する場合は、環境変数 PATH に Java 実行コマンドがあるディレクトリを追加してください。
- 全文検索用データ変換定義ファイルの文字コードは、TS2XMLcnv コマンド実行時のロケールと同じ文字コードにしてください。
- XML 文書の解析エラーが発生した場合でも処理を続行し、変換可能なファイルについては全文検索用データファイルを作成します。解析エラーが発生した場所は、エラー出力ファイルを参照して特定します。
- 続行不可能なエラーが発生した場合は処理を中断し、カレントディレクトリに「ERR\_TRACE\_年\_月\_日」の名称でエラートレースファイルを出力します。同じ日に発生したエラーは、同じファイルに追記します。
- Windows 上で TS2XMLcnv コマンド引数のパス名に空白を含めたい場合は、前後を引用符 (") で囲んでください。

## (9) 環境設定

Windows 上で TS2XMLcnv コマンドを実行する場合は、次の環境変数を設定する必要があります。

表 D-1 Windows 上で TS2XMLcnv コマンドを実行する場合に設定する環境変数

環境変数名	指定値
TS2_COSMINEXUS_PATH	Cosminexus インストールフォルダ
TS2_BIB2_LIBRARY_PATH	Library インストールフォルダ

## (10) Cosminexus Version 7 以降を使用する場合

使用する Cosminexus のバージョンが 7 以降の場合、ユーザー定義ファイルの作成、および環境変数の設定を実行する必要があります。ここでは設定方法について説明します。指定値の詳細についてはご使用になる Cosminexus のバージョンにより、次のどれかのマニュアルを参照してください。

「Cosminexus Version 7 リファレンス定義編」の「EJB クライアントアプリケーションで使用するファイル」

「Cosminexus Version 7 リファレンス定義編」の「Java アプリケーションで使用するファイル」

「Cosminexus Version 8 リファレンス定義編 (サーバー定義)」の「Java アプリケーションで使用するファイル」

なお、ユーザー定義ファイルの作成、および環境変数の設定を省略した場合でも、TS2XMLcnv コマンドは動作しますが、Java のトラブルシュート用情報が出力されないため、問題が発生したときの解決が困難になるおそれがあります。

環境変数名



## TS2\_XMLCNV\_CONFPATH

## 指定値

ユーザー定義ファイル格納ディレクトリパスを指定します。ユーザー定義ファイルの名称を次に示します。

- usrconf.cfg
- usrconf.properties

## usrconf.cfg ファイルの定義内容

```

ejb.client.log.directory=<log_directory_base>
ejb.client.ejb.log=<log_directory_name>
ejb.client.log.stdout.enabled=false
add.jvm.arg=-XX:-HitachiOutOfMemoryAbort

```

## usrconf.cfg ファイル定義時の注意事項

- <log\_directory\_base> , および <log\_directory\_name> 以外は固定です。
- <log\_directory\_base> には、ログファイルの出力先ディレクトリを絶対パスで指定してください。
- <log\_directory\_name> には、`ejb.client.log.directory` キーに指定したディレクトリ下の Java アプリケーションが出力する、メッセージログ出力先ディレクトリを指定してください。
- ログファイルは、<log\_directory\_base> 指定ディレクトリの <log\_directory\_name> 指定ファイルに出力されます。
- <log\_directory\_base> , および <log\_directory\_name> に指定するディレクトリには、TS2XMLcnv コマンドを実行するユーザーのアクセス権が必要です。
- TS2XMLcnv コマンドを複数同時に実行する場合は、それぞれログ出力が異なる内容のユーザー定義ファイルを作成し、TS2\_XMLCNV\_CONFPATH 設定値を各定義ファイル格納ディレクトリパスに設定してください。

## usrconf.properties ファイルの定義内容

usrconf.properties ファイルではログファイルの出力量を指定できますが、デフォルトの値で十分な情報が取得できる設定となっています。

デフォルトで TS2XMLcnv コマンドを実行する場合は、usrconf.properties ファイルのサイズは 0 バイトで作成してください。

デフォルトの設定ではログファイル出力量は数十メガバイトありますが、出力量を変更する場合は、次の定義を記述してください。

```

ejbserver.logger.channels.define.<チャンネル名>.filenum=<file_num>
ejbserver.logger.channels.define.<チャンネル名>
>.filesize=<file_size>

```

## usrconf.properties ファイル定義時の注意事項

- <file\_num> には、ログファイル面数を指定してください。
- <file\_size> には、1 面当たりのログファイルサイズを指定してください。
- <チャンネル名> には次の名称を指定できます。なお、指定できる名称以外を指定した場合、正しく動作しないおそれがあります。

ClientMessageLogFile , ClientExceptionLogFile , ClientMaintenanceLogFile ,  
EJBContainerLogFile , UserOutLogFile , UserErrLogFile

## (11) コマンドが出力するエラーメッセージ

ここでは、TS2XMLenv コマンドが出力するエラーメッセージについて説明します。

### (a) エラーメッセージの形式

TS2XMLenv コマンドが出力するエラーメッセージは、メッセージ ID とそれに続くテキストから構成されます。メッセージの形式を次に示します。

KIBSnnnn - X メッセージテキスト

ただし、TS サーバーのメッセージがイベントログに出力される場合は、メッセージの番号だけがイベント ID になります。このとき「KIBS」やメッセージの種類は出力されません。

KIBS

メッセージであることを示します。

nnnn

メッセージの番号です。4 けたの固有の番号が付いています。

X

メッセージの種類を表します。次にその意味を示します。

I

システムの動作を通知します。

W

処理は続行しますが、障害が発生したので通知します。

E

障害が発生したので、処理を中断します。

メッセージテキスト

出力されるメッセージのテキスト（内容）です。なお、メッセージテキストの「XXX」などの文字は、メッセージが出力される状況によって変わる値です。説明が必要なものについては、メッセージテキストの後ろにその内容を示します。

要因

エラーが発生した原因を説明しています。

対処

エラーを解除する対策について説明しています。なお、「保守員に連絡してください」とは、購入時の契約に基づいて、システム管理者が弊社問い合わせ窓口へ連絡することを示します。

(b) エラーメッセージの詳細

**KIBS0031-I**

---

全文検索用データファイルを作成しました。

要因

全文検索用データファイルの作成を終了しました。

**KIBS1000-W**

---

文書登録前処理でエラーが発生しました。

要因

検索対象ファイルの文書登録前処理でエラーが発生しました。

対処

エラー出力ファイルの内容を参照して、エラーになった項目を修正して再実行してください。

**KIBS2016-E**

---

ファイル (XX) がオープンできません。(YY)

要因

XX に示されるファイルのオープンに失敗しました。

XX：オープンできなかったファイル名

YY：詳細情報

対処

詳細情報を基に、エラーになった原因を取り除いて再実行してください。

**KIBS2024-E**

---

コマンド引数の指定に誤りがあります。

要因

コマンドの引数に誤りがあります。または、必ず指定する引数が指定されていません。

対処

引数の指定を見直して、再実行してください。

**KIBS2087-E**

---

全文検索用データ変換定義ファイルの指定に誤りがあります。(XX)

要因

全文検索用データ変換定義ファイルの指定に誤りがあるため、XML の変換処理を実行できません。

XX：エラーの発生した行番号

対処

全文検索用データ変換定義ファイルの指定を見直して、再度実行してください。

### **KIBS2088-E**

---

環境変数 XX が不正です。

要因

指定された環境変数が設定されていないか、指定値が不正なためコマンドを実行できません。

XX：不正な環境変数

対処

マニュアルを参照し、指定された環境変数に正しい値を指定してください。

### **KIBS9000-E**

---

MEMORY SHORTAGE(XX,YY)

要因

処理に必要なメモリをシステムが取得できませんでした。

XX：メモリを要求したソースファイル名

YY：エラーが発生した行

対処

サーバーのメモリの状態、または TS2XMLCNV\_MAXMEM の値を確認してください。

### **KIBS9900-E**

---

FILE IO ERROR(VV,WW,XX,YY,ZZ)

要因

ファイルのアクセス時にエラーを検出しました。

VV：ファイル名

WW：オペレーション

XX：詳細情報

YY：オペレーションを発行したソースファイル

ZZ：オペレーションを発行した行

対処

保守員に連絡してください。

### **KIBS9902-E**

---

SYSTEM ERROR(XX,YY,ZZ)

要因

サーバー内部で自己矛盾を検出しました。

XX：詳細情報

YY：エラーの発生したファイル

ZZ：エラーの発生した行

対処

保守員に連絡してください。

## 付録 E 関数のエラーステータス

エラーステータスは、Development Kit または Server Library が提供する関数が異常終了したときに戻り値として出力されます。ここでは、関数のエラーステータスについて説明します。

### 付録 E.1 エラーステータスの分類

エラーステータスは、次の表に示す範囲で分類されます。

表 E-1 エラーステータスの範囲と意味

エラーステータス	意味
-110000 ~ -119999	Development Kit または Server Library 内で発生したエラー
-120000 ~ -129999	サーバーとの通信処理で発生したエラー
-130000 ~ -139999	Development Kit または Server Library で提供する関数の実行中に発生したエラー
-200000 ~ -219999	サーバー内で発生したエラー

エラーステータスのマクロ名称は、すべてインクルードファイル TS2lib.h で宣言されています。

### 付録 E.2 エラーステータスの形式

エラーステータスを次に示す形式で説明します。

#### エラーステータス

出力されるエラーステータスです。

#### エラーマクロ

出力されるエラーマクロです。

#### 内容

エラーの内容を説明しています。

#### 要因

エラーが発生した原因を説明しています。

#### 対処

エラーを解除する対策について説明しています。なお、「保守員に連絡してください」とは、購入時の契約に基づいて、システム管理者が弊社問い合わせ窓口へ連絡することを示します。

## 付録 E.3 エラーステータスの詳細

### -110000

---

TS2DERL\_API\_SYSTEM

内容

システムエラーが発生しました。

要因

関数の実行中に回避できない異常が発生しました。

対処

保守員に連絡してください。

### -110002

---

TS2DERL\_ALOC\_MEM

内容

メモリが確保できませんでした。

要因

関数を実行中に空きメモリがなくなりました。

対処

実行中のほかのプロセスを終了して再実行してください。

### -120001

---

TS2DERL\_GET\_PORT

内容

ポート番号が取得できませんでした。

要因

services ファイルに Bibliotheca2 TS のポート番号が登録されていませんでした。

対処

services ファイルに Bibliotheca2 TS のポート番号を登録して再実行してください。

### -120002

---

TS2DERL\_ABEND\_COM

内容

サーバーとの通信処理中に異常が発生しました（続いて関数名とエラーコードが出力されます）。

要因

- 通信回線に異常が発生しました。
- サーバーに異常が発生しました。
- サーバーでセッションタイムアウトが発生した可能性があります。

対処

- 通信経路を確認して再実行してください。
- サーバーが起動しているかを確認して再実行してください。
- 関数名で「(connect,XXX)」が表示される場合は、サーバーの通信リソースが枯渇しています。しばらく時間を置いてから通信を再開してください。  
XXX：任意の半角数字
- TS2F\_con 関数でサーバーに再接続してください。

---

**-120003**

TS2DERL\_OVER\_TIME

内容

タイムアウトが発生したため処理を中断しました。

要因

サーバーでの処理時間が規定時間を超えたため処理を中断しました。

対処

サーバーが起動しているかを確認して再実行してください。

---

**-120004**

TS2DERL\_GET\_SOCKET

内容

ソケットが確保できませんでした（続いてエラーコードが出力されます）。

要因

使用できるソケットがありませんでした。

対処

ソケットまたはファイルを使用しているほかのプロセスを終了して再実行してください。

---

**-120007**

TS2DERL\_CANT\_EXEC

内容

機能が正しくインストールされていないため、実行できません。

要因

Development Kit , Runtime Library , または Server Library が正しくインストールされていません。

対処

Development Kit , Runtime Library , または Server Library を正しくインストールしてください。



**-130001**

---

## TS2DERL\_NLPTR\_SERVER

## 内容

サーバー名格納領域のポインタが NULL です。

## 要因

サーバー名のポインタに NULL が指定されています。

## 対処

正しいポインタを指定してください。

**-130002**

---

## TS2DERL\_BADARG\_SERVER

## 内容

指定されたサーバー名は登録されていませんでした。

## 要因

hosts ファイルに登録されていないホスト名が指定されています。

## 対処

hosts ファイルにホスト名を登録して再実行してください。

**-130003**

---

## TS2DERL\_BADARG\_SESSIONID

## 内容

セッション識別子が不正です。

## 要因

- 不正なセッション識別子が指定されています。
- サーバーでセッションタイムアウトが発生した可能性があります。

## 対処

- TS2F\_con 関数で返された正しいセッション識別子を指定してください。
- TS2F\_con 関数でサーバーに再接続してください。

**-130004**

---

## TS2DERL\_NLPTR\_CID

## 内容

コマンド ID 格納領域のポインタが NULL です。

## 要因

コマンド ID のポインタに NULL が指定されています。

## 対処

正しいポインタを指定してください。

## **-130005**

---

### TS2DERL\_BADARG\_CID

内容

コマンド ID の値が不正です。

要因

コマンド ID に不正な値が指定されています。

対処

TS2F\_cmd 関数または TS2F\_sndcmd 関数で返された正しいコマンド ID を指定してください。

## **-130006**

---

### TS2DERL\_NLPTR\_CMD

内容

サーチコマンド格納領域のポインタが NULL です。

要因

サーチコマンド格納領域のポインタに NULL が指定されています。

対処

正しいポインタを指定してください。

## **-130009**

---

### TS2DERL\_CANTEX\_CID

内容

指定されたコマンドはキャンセルできません。

要因

- 実行が終了したサーチコマンドをキャンセルしようとしてしました。
- 実行が終わっていないサーチコマンドの結果を破棄しようとしてしました。

対処

- 実行結果を受け取るかまたは破棄してください。
- サーチコマンドの実行が終わってから再実行してください。

## **-130010**

---

### TS2DERL\_CCLEL\_CID

内容

指定されたサーチコマンドはキャンセルされました。

要因

- 指定されたサーチコマンドは TS2F\_cancelcmd 関数によってキャンセルされました。

- 指定されたサーチコマンドは TS2F\_canceldat 関数によって実行結果が破棄されました。

対処  
なし

---

**-130011**

---

**TS2DERL\_NLPTR\_RBUF**

内容  
実行結果格納領域のポインタが NULL です。

要因  
実行結果格納領域のポインタに NULL が指定されています。

対処  
正しいポインタを指定してください。

---

**-130012**

---

**TS2DERL\_BADARG\_RSIZ**

内容  
実行結果格納領域サイズの値が不正です。

要因

- 実行結果格納領域のサイズに負の値が指定されています。
- 実行結果格納領域のポインタは NULL ですが、実行結果格納領域サイズに 0 以外のバイト数が指定されています。
- 実行結果格納領域のサイズは 0 ですが、実行結果格納領域のポインタに NULL 以外の値が指定されています。

対処  
正しい領域サイズを指定してください。

---

**-130014**

---

**TS2DERL\_BADARG\_MODE**

内容  
mode の値が不正です。

要因  
mode に不正な値が指定されています。

対処  
正しいモードのマクロを指定してください。

---

**-130015**

---

**TS2DERL\_CANTEX\_MODE**

内容

指定の mode では実行できません。

要因

実行結果不足エラーが発生していないサーチコマンドに対して、TS2DL\_RETRY  
モードで実行しようとした。

対処

関数発行のシーケンスに誤りがないかを確認してください。

---

**-130016**

TS2DERL\_LESS\_BUFSIZ

内容

実行結果格納領域のサイズが不足しています。

要因

実行結果格納に必要な領域が確保されていません。

対処

- 結果格納領域を拡張した上で、再実行してください。
- TS2F\_canceldat 関数によってサーチコマンドの実行結果を破棄してください。

---

**-130017**

TS2DERL\_NLPTR\_MLOOK

内容

サーチコマンド MLOOK の実行結果のポインタが NULL です。

要因

サーチコマンド MLOOK の実行結果へのポインタに NULL が指定されています。

対処

正しいポインタを指定してください。

---

**-130018**

TS2DERL\_NLPTR\_DOC

内容

文書のポインタが NULL です。

要因

文書のポインタに NULL が指定されています。

対処

文書が格納されている領域を指定してください。

---

**-130019**

TS2DERL\_BADARG\_DOC

## 内容

指定された文書が不正です。

## 要因

指定された文書が不正です。

## 対処

サーチコマンド MDISPLAY で取得した文書を指定してください。

**-130020**

---

## TS2DERL\_BADARG\_MLOOK

## 内容

サーチコマンド MLOOK の実行結果が不正です。

## 要因

不正なサーチコマンド MLOOK の実行結果が指定されています。

## 対処

正しいサーチコマンド MLOOK の実行結果を指定してください。

**-130023**

---

## TS2DERL\_BADARG\_INDTSIZ

## 内容

入力データが不正です。

## 要因

TS2F\_cmd\_DATA 関数または TS2F\_sndcmd\_DATA 関数に指定した入力データ長が不正です。

## 対処

正しい値を指定してください。

**-130025**

---

## TS2DERL\_NLPTR\_INDTBUF

## 内容

入力データポインタが NULL です。

## 要因

TS2F\_cmd\_DATA 関数または TS2F\_sndcmd\_DATA 関数に指定した入力データポインタが NULL です。

## 対処

正しいポインタを指定してください。

### **-130034**

---

#### TS2DERL\_CMD\_TOOLONG

内容

サーチコマンドの文字列が長過ぎます。

要因

サーチコマンドの文字列が長過ぎます。

対処

サーチコマンドの文字列を確認してください。

### **-130035**

---

#### TS2DERL\_NOTEND

内容

サーチコマンドが終了していません。

要因

サーチコマンドが終了していません。

対処

サーチコマンドの終了を待ってください。

### **-130038**

---

#### TS2DERL\_CANT\_GET\_CID

内容

同時に実行できるコマンド数を超えました。

要因

サーバーが同時に実行できるコマンド数を超えて、コマンドを実行しようとした。

対処

実行中のコマンドを終了させてから再実行してください。

### **-130042**

---

#### TS2DERL\_VER\_UNMATCH

内容

ライブラリのバージョンが不正です ( stlib : XX , lib : YY )

要因

アプリケーションのリンケージに使用したライブラリと、実行時にロードするライブラリのバージョンが不正です。

XX : 保守情報です。

YY : 保守情報です。

対処

アプリケーションのリンケージに使用したライブラリと、実行時にロードするライブラリのバージョンを合わせてください。

**-130045**

---

TS2DERL\_BADARG\_XMLNUM

内容

XML 数が不正です。

要因

入力 XML 情報数に不正な値が指定されています。

対処

入力 XML 情報数の指定を確認してください。

**-130055**

---

TS2DERL\_BADARG\_XMLOPT

内容

XML 変換オプションのポインタが NULL です。

要因

XML 変換オプションのポインタに NULL が指定されています。

対処

正しいポインタを指定してください。

**-130056**

---

TS2DERL\_BADARG\_CPLMODE

内容

変換結果のデータ種別が不正です。

要因

変換結果のデータ種別に不正な値が指定されています。

対処

正しい変換結果のデータ種別の値を指定してください。

**-130057**

---

TS2DERL\_BADARG\_CHARSPEC

内容

出力文字コード仕様が不正です。

要因

出力文字コード仕様に不正な値が指定されています。

対処

正しい出力文字コード仕様の値を指定してください。

### **-130058**

---

TS2DERL\_BADARG\_XMLDAT

内容

XML データ情報のポインタが NULL です。

要因

XML データ情報のポインタに NULL が指定されています。

対処

正しいポインタを指定してください。

### **-130059**

---

TS2DERL\_BADARG\_XMLENTRY

内容

XML データエントリが不正です。

要因

文書 ID , XML 文書データサイズ , XML 文書データ格納ポインタのどれかが不正です。

対処

XML データエントリの指定を確認してください。

### **-130060**

---

TS2DERL\_BADARG\_RESISB

内容

変換結果格納領域の受け取り領域ポインタが NULL です。

要因

全文検索用データ変換結果格納領域の受け取り領域ポインタに NULL が指定されています。

対処

正しいポインタを指定してください。

### **-130061**

---

TS2DERL\_BADARG\_RESISBLL

内容

変換結果サイズの受け取り領域ポインタが NULL です。

要因

全文検索用データ変換結果サイズの受け取り領域ポインタに NULL が指定されてい



ます。

対処

正しいポインタを指定してください。

### **-130062**

---

TS2DERL\_OVER\_MAXSIZ

内容

変換結果サイズがシステムの上限值を超えました ( 続いて変換結果のサイズが出力されず )。

要因

変換結果サイズがシステムの上限值を超えました。

対処

XML 文書データのサイズ、または変換結果のデータ種別を見直してください。

### **-130063**

---

TS2DERL\_WF\_XMLPARSE\_ERR

内容

XML パーサでエラーが発生しました ( 続いて詳細エラー情報が出力されず )。

要因

XML パーサでエラーが発生しました。

対処

詳細エラー情報の内容を確認してください。

### **-200001**

---

TS2DERL\_ERR\_IN\_SERVER

内容

サーバーでエラーが発生しました。

要因

サーバーでエラーが発生しました。

対処

TS2F\_geterror 関数で詳細エラー情報を取得してください。

---

## 付録 F TS サーバーのエラーコード

TS サーバーのエラーコードは、サーチコマンドを実行してエラーが発生した場合に出力されます。ここでは、TS サーバーのエラーコードについて説明します。

### 付録 F.1 エラーコードの形式

TS サーバーのエラーコードを次に示す形式で説明します。

#### エラーコード

出力されるエラーコードです。

#### メッセージテキスト

エラーコードに対するメッセージの内容です。メッセージテキストは、TS2F\_geterror 関数で取得できます。ただし、エラーコードが 0000 の内容は取得できません。

#### 要因

エラーが発生した原因を説明しています。

#### 対処

エラーを解除する対策について説明しています。なお、「保守員に連絡してください」とは、購入時の契約に基づいて、システム管理者が弊社問い合わせ窓口へ連絡することを示します。

### 付録 F.2 エラーコードの詳細

#### 0000

---

エラーは発生していません。

#### 要因

エラーは発生していません。

#### 2000

---

バージョン番号が不正です。

#### 要因

サーバーのバージョンと、クライアントのバージョンが一致していません。

#### 対処

サーバーのバージョンとクライアントのバージョンを確認してください。

#### 2001

---

指定されたテキスト DB(XX) は存在しません。

**要因**

指定されたテキストデータベースは存在しません。  
XX：テキストデータベース名

**対処**

テキストデータベース名の指定を確認してください。

**2002**

---

指定されたテキスト DB(XX) は既に存在します。

**要因**

指定されたテキストデータベース名は、すでに存在しています。  
XX：テキストデータベース名

**対処**

テキストデータベース名の指定を確認してください。

**2003**

---

テキスト DB(XX) はほかの処理で使用中です。

**要因**

指定したテキストデータベースは、ほかの処理が使用しているため使用できません。  
XX：テキストデータベース名

**対処**

ほかの処理が終了するのを待って再実行してください。

**2004**

---

作成できるテキスト DB の数を超えています。

**要因**

作成できるテキストデータベースの数を超えているため、テキストデータベースを作成できません。

**対処**

不要なテキストデータベースを削除して、再実行してください。

**2005**

---

文書 ID の指定に誤りがあります。

**要因**

文書 ID に不正な値が指定されたか、または文書 ID が指定されていませんでした。

**対処**

文書 ID に指定した値を確認してください。

## 2006

---

条件式の指定に誤りがあります。(XX)

要因

指定された検索条件に誤りがありました。

XX：エラー詳細

対処

条件式の指定を見直し、再実行してください。

## 2007

---

テキスト DB ディレクトリ (XX) の指定に誤りがあります。

要因

テキスト DB ディレクトリに指定されたディレクトリ名の長さが上限を超えています。または、存在しないパス名が指定されています。

XX：テキスト DB ディレクトリ名

対処

テキスト DB ディレクトリに指定した値を確認してください。

## 2008

---

指定されたディレクトリ (XX) にはファイルがあります。

要因

指定されたディレクトリにはファイルまたはディレクトリがあります。

XX：ディレクトリ名

対処

ディレクトリに指定した値を確認してください。

## 2009

---

テキスト DB 名 (XX) の指定に誤りがあります。

要因

テキストデータベース名の指定に誤りがあります。

XX：テキストデータベース名

対処

テキストデータベース名の指定を確認してください。

## 2010

---

指定されたパス名 (XX) の指定に誤りがあります。

要因

指定されたパスはありません。または指定されたパス名の長さは上限を超えていま

す。  
XX：パス名

対処  
指定したパス名を確認してください。

## 2011

---

文書の長さが上限を超えています。

要因  
文書の長さが上限を超えています。

対処  
指定したテキストデータを見直し、再実行してください。

## 2012

---

辞書ソースファイルの内容に誤りがあります。(LINE:XX,YY)

要因  
同義語辞書の作成時に指定された辞書ソースファイルの内容に誤りがあります。  
XX：エラーの発生した行  
YY：詳細エラー

対処  
辞書ソースファイルの内容を確認してください。

## 2013

---

処理の実行がキャンセルされました。

要因  
キャンセル指示によって、処理がキャンセルされました。

## 2014

---

テキスト DB の最大容量を超えるため、文書を登録できませんでした。(XX)

要因  
テキストデータベースの最大容量を超えるため、文書を登録できませんでした。  
XX：最大容量を超えた物理格納定義の種別

対処  
新しい物理格納ディレクトリを追加してください。

## 2015

---

登録用ワークエリアが不足したため、文書を登録できませんでした。

要因  
登録用ワークエリアが不足したため、文書を登録できませんでした。

対処

登録する文書の長さまたは文書数を減らして、再実行してください。

## 2016

---

ファイル (XX) がオープンできません。(YY)

要因

XX に示されるファイルのオープンに失敗しました。

XX：オープンできなかったファイル名

YY：詳細情報

対処

詳細情報を基に、エラーになった原因を取り除いて再実行してください。

## 2017

---

ディレクトリ (XX) を作成できませんでした。

要因

ディレクトリの作成時にエラーが発生しました。

XX：ディレクトリ名

対処

続いて出力される詳細エラーを基に原因を取り除き、再実行してください。

## 2018

---

ディレクトリ名又はファイル名はドライブ名からのフルパス名で指定してください。

要因

指定されたディレクトリ名またはファイル名がフルパスで指定されていません。

対処

ディレクトリ名またはファイル名は、ドライブ名からのフルパスで指定してください。

## 2019

---

XX の指定に誤りがあります。

要因

XX に示される項目の指定値に誤りがあります。

XX：指定に誤りがあった項目

対処

正しい値を指定してください。

## 2020

---

XX が入力されていません。

**要因**

XX に示される項目に値が入力されていません。

XX : 入力が必要な項目

**対処**

正しい値を入力してください。

**2021**

---

文書 ID(XX) は既に存在します。

**要因**

指定された文書 ID はすでに存在します。

XX : 文書 ID

**対処**

文書 ID を確認してください。

**2022**

---

同義語辞書が既に作成されています。

**要因**

テキストデータベースには同義語辞書がすでに作成されています。

**対処**

同義語辞書を再作成する場合は、すでに作成されている同義語辞書を削除してから作成してください。

**2023**

---

同義語辞書が作成されていません。

**要因**

テキストデータベースには同義語辞書が作成されていません。

**対処**

同義語辞書を削除するテキストデータベースの名称が正しいかを確認してください。

**2024**

---

コマンド引数の指定に誤りがあります。

**要因**

コマンド引数に誤りがあります。または、必ず指定する引数が指定されていません。

**対処**

引数の指定を見直して、再実行してください。

**2025**

---

サーバーが起動されていません。

要因

サーバーが起動されていません。

対処

サーバーを起動してください。

## 2026

---

テキスト D B (XX) は既にオープンされています。

要因

すでにオープンしているテキストデータベースをオープンしようとしてしました。

XX：テキストデータベース名

対処

テキストデータベースの状態を確認して、再実行してください。

## 2027

---

テキスト D B (XX) は処理中のため、クローズできません。

要因

処理中のテキストデータベースをクローズしようとしてしました。

XX：テキストデータベース名

対処

テキストデータベースに対する処理が終了するのを待って、再実行してください。

## 2028

---

検索処理で使用可能なメモリ量を超えたため、検索できません。(XX)

要因

使用できるメモリ量を超えたため、検索できませんでした。

XX：指定された使用可能なメモリ量

対処

検索に使用できるメモリ量を増やしてください。または、検索条件を変更し、検索に使用できるメモリ量を超えないように指定してください。

## 2029

---

指定された検索結果集合は存在しません。

要因

指定された検索結果集合は存在しないか、または削除されています。

対処

指定した検索結果集合 ID を確認してください。



**2030**

---

テキスト DB(XX) は構造文書用テキスト DB ではありません。

**要因**

指定されたテキストデータベースが構造文書用テキストデータベースではないため、処理が実行できませんでした。

XX：テキストデータベース名

**対処**

テキストデータベースの指定を見直して、再実行してください。

**2031**

---

テキスト DB(XX) はオープンされていません。

**要因**

サーチコマンド MSTART の実行時、指定されたテキストデータベースがオープンされていませんでした。

XX：テキストデータベース名

**対処**

テキストデータベースをオープンして、再実行してください。

**2032**

---

ファイル (XX) は検索対象ファイルではありません。

**要因**

指定されたファイル名は検索対象ファイルではなかったため、処理できませんでした。

XX：指定されたファイル名

**対処**

ファイル名の指定を見直して、再実行してください。

**2033**

---

ファイル (XX) は、テキスト DB に定義されていません。

**要因**

指定されたファイルは、テキストデータベースに定義されていませんでした。

XX：指定されたファイル名

**対処**

ファイルの指定を見直して、再実行してください。

**2034**

---

既に MSTART が実行されています。

**要因**

この検索セッションでは、すでにサーチコマンド MSTART が実行されています。

**対処**

サーチコマンド MEND を実行して、データベース使用宣言を終了させてください。  
そのあと、サーチコマンド MSTART を実行するかどうかを確認してください。

---

## 2035

MSTART が実行されていません。

**要因**

サーチコマンド MSTART を実行しておく必要のあるサーチコマンドの実行時、サーチコマンド MSTART が実行されていませんでした。

**対処**

サーチコマンド MSTART を実行して、再実行してください。

---

## 2036

同義語辞書が設定されていません。

**要因**

このテキストデータベースには、同義語辞書が設定されていないため、同義語展開できません。

**対処**

テキストデータベースに同義語辞書を設定して、再実行してください。

---

## 2037

サーチコマンドのパラメーター (XX) に誤りがあります。

**要因**

サーチコマンドのパラメーターに誤りがありました。構文エラーや、パラメーターの範囲が指定できる範囲を超えているものも含まれます。

XX：誤りがあったパラメーター。どのパラメーターにエラーがあったのか不明な場合は、\* が出力されます。

**対処**

パラメーターの指定を見直して、再実行してください。

---

## 2038

サーチコマンドに必要な入力データが設定されていません。

**要因**

サーチコマンドに入力データが設定されていませんでした。

**対処**

必要な入力データを設定して、再実行してください。

**2039**

---

サーチコマンドの指定に誤りがあります。

**要因**

誤ったサーチコマンドが指定されたか、またはサーチコマンドが指定されていません。

**対処**

サーチコマンドの指定を見直して、再実行してください。

**2040**

---

検索結果集合を作成できません。

**要因**

検索結果集合 ID がシステムで作成できる上限値を超えたため、検索結果集合を作成できませんでした。

**対処**

パラメーターを指定しないサーチコマンド MSWEEP を実行して、検索結果集合をすべて削除してから再実行してください。

**2041**

---

ディレクトリ名 (XX) の指定に誤りがあります。

**要因**

指定されたディレクトリ名に誤りがあります。  
XX : ディレクトリ名

**対処**

ディレクトリ名の指定を見直して、再実行してください。

**2042**

---

ファイル (XX) は存在しません。

**要因**

指定されたファイルは Windows または UNIX 上に存在しません。  
XX : ファイル名

**対処**

指定したファイル名を見直して、再実行してください。

**2043**

---

検索対象ファイル (XX) の指定に誤りがあります。

**要因**

検索対象ファイルの指定に誤りがあります。

XX：検索対象ファイル名

対処

検索対象ファイルの指定を見直して、再実行してください。

## 2044

---

関連情報ファイル (XX) の指定に誤りがあります。

要因

関連情報ファイルの指定に誤りがあります。

XX：関連情報ファイル名

対処

関連情報ファイルの指定を見直して、再実行してください。

## 2045

---

指定された物理格納ディレクトリ名は、既に定義で使用されています。

要因

物理格納ディレクトリに指定されたディレクトリ名は、すでにデータベース定義で使用されています。

対処

物理格納ディレクトリの指定を見直して、再実行してください。

## 2046

---

初期容量と最大容量の値が異なります。

要因

増分容量に 0 が指定されていて、増分されていないにもかかわらず、初期容量と最大容量の値が異なります。

対処

容量の指定を見直してください。

## 2047

---

指定された値では増分できません。

要因

初期容量と増分容量の和が最大容量の値を超えるため、増分できません。

対処

容量の指定を見直してください。

## 2048

---

最大容量を超える値が初期容量に指定されています。

要因

初期容量に最大容量を超える値が指定されています。

対処

容量の指定を見直してください。

---

## 2049

定義可能な最大容量を超える最大容量値が指定されています。

要因

最大値を超える値が最大容量として指定されています。

対処

容量の指定を見直して、残物理格納ファイル定義可能サイズ以下の容量にしてください。なお、Bibliotheca2 TextSearch Extension Version 2 を組み込むことで、使用できるテキストデータベースの容量（残物理格納ファイル定義可能サイズ）を増やせます。

---

## 2050

サーバーの停止処理中です。

要因

サーバーが停止処理中のため、処理を実行できません。

対処

サーバーの再起動後、再実行してください。

---

## 2051

接続中のセッションが上限を超えています。

要因

接続中のセッション数が上限を超えているため、接続できません。

対処

ほかのセッションが解放されるのを待って、再実行してください。

---

## 2052

セッション識別子が不正です。

要因

指定されたセッション識別子は存在しません。

対処

サーバーを再起動していないかを確認してください。または、クライアントプログラムの処理を見直してください。

---

## 2053

コマンド実行中です。

要因

コマンドの実行中のため、コマンドの実行要求は拒否されました。

対処

実行中のコマンドが終了するのを待って、再実行してください。

---

## 2054

コマンドの実行権限がありません。

要因

使用しているクライアントには、このコマンドを実行できる権限がありません。

対処

クライアントプログラムを見直してください。

---

## 2055

このクライアントでは、この機能を利用できません。

要因

接続されているクライアントでは、この機能は利用できません。

対処

接続しているクライアントを見直してください。

---

## 2056

ベースとなる検索結果集合が存在しません。

要因

ベースとなる検索結果集合が存在しないか、または削除されています。

対処

ベースとして指定されている検索結果集合 ID を確認してください。

---

## 2057

新しい検索結果集合を作成できませんでした。

要因

システムで管理している情報ファイルの上限に達したため、検索結果集合を作成できませんでした。

対処

サーチコマンド MSWEEP で検索結果集合を削除してください。

---

## 2058

最大検索結果集合 ID 以上の値が指定されました。

要因

検索結果集合の削除時に、検索結果集合 ID の最大値を超える値が指定されました。

## 対処

検索結果集合 ID の指定を見直してください。

**2059**

---

異表記展開された結果の語数が上限値を超えました。

## 要因

異表記展開の結果、展開された語数が上限値を超えました。

## 対処

検索タームおよび異表記展開の指定を見直して、再実行してください。

**2060**

---

ランキング情報がありません。

## 要因

ランキング情報がありませんでした。

## 対処

ランキング検索してランキング情報を作成後、再実行してください。

**2061**

---

登録文書情報定義ファイルの指定に誤りがあります。(XX)

## 要因

登録文書情報定義ファイルに誤りがあるため、文書登録前処理または関連情報の登録処理ができませんでした。誤りの内容は次のとおりです。

- 文書 ID の指定値が不正である
- 登録文書情報定義ファイルの指定がない

XX：エラーの発生した行番号

## 対処

登録文書情報定義ファイルの内容を確認してください。

**2062**

---

文書登録前処理が行われていません。

## 要因

文書登録前処理を実行していない文書を登録しようとしてしました。

## 対処

文書登録前処理を実行してから、文書を登録してください。

**2063**

---

検索結果集合 ID が上限に達しているため、処理を実行できません。

要因

すでに作成されている検索結果集合 ID の最大値がサーバー設定ファイルに設定された検索結果集合 ID の上限値に達しているため、処理を実行できませんでした。

対処

サーチコマンド MSWEEP ですべての検索結果集合を削除して、再実行してください。

---

## 2064

文書登録前処理が実行されていません。

要因

文書登録前処理が実行されていないため、文書を登録できませんでした。

対処

文書登録前処理を実行して、再実行してください。

---

## 2065

削除できる文書数を超えています。

要因

削除する文書数が多過ぎるため、文書を削除できませんでした。

対処

複数回に分けて文書を削除してください。

---

## 2066

DB ユティリティは既に起動しています。

要因

すでに DB ユティリティが起動している状態で、DB ユティリティを起動しました。

対処

すでに起動している DB ユティリティで処理を実行してください。

---

## 2067

文書が登録されていません。

要因

文書の検索時に、検索対象ファイルおよび関連情報ファイルに文書が 1 件も登録されていませんでした。

対処

検索対象ファイルおよび関連情報ファイルに文書を登録して、再実行してください。

---

## 2068

実行権限がないため、処理できません。



**要因**

指定された処理を実行するための権限がないため、実行できませんでした。

**対処**

管理者権限を持つユーザーが処理を実行してください。

**2069**

---

残物理格納ファイル定義可能サイズが 0 のため、処理できません。

**要因**

テキストデータベースを作成または再構成しようとしたが、残物理格納ファイル定義可能サイズが 0 のため、処理を実行できませんでした。

**対処**

不要なテキストデータベースを削除して、残物理格納ファイル定義可能サイズを増やして再実行してください。なお、Bibliotheca2 TextSearch Extension Version 2 を組み込むことで、使用できるテキストデータベースの容量（残物理格納ファイル定義可能サイズ）を増やせます。

**2070**

---

定義できる物理格納ディレクトリ数を超えています。

**要因**

定義できる物理格納ディレクトリ数を超えているため、追加できませんでした。

**対処**

テキストデータベースのディレクトリ構成を見直して、物理格納ディレクトリ数が上限を超えないよう再作成してください。

**2071**

---

テキスト DB コンデンス実行中のため処理できません。

**要因**

テキストデータベースのコンデンスを実行中のため、サーチコマンドは実行できません。

**対処**

テキストデータベースのコンデンス終了後、再実行してください。

**2072**

---

サーバー内の管理テーブルに空きがないため処理できません。

**要因**

サーバー内の管理テーブルに空きがないため、処理の実行要求は拒否されました。

**対処**

ほかの実行中の処理が終了するのを待って、再実行してください。しばらく待っても実行できない場合は、サーバーを再起動してください。

## 2073

---

使用できないテキスト DB です。

### 要因

- クライアントが一般の場合  
Web Search 用または Groupmax Document Manager 用のテキストデータベースを使用しようとしてしました。
- クライアントが Web Search または Groupmax Document Manager の場合  
Web Search または Groupmax Document Manager のテキストデータベースの指定が正常ではありません。

### 対処

- クライアントが一般の場合  
指定したテキストデータベース名を見直してください。
- クライアントが Web Search または Groupmax Document Manager の場合  
Web Search または Groupmax Document Manager のテキストデータベース指定を見直してください。

## 2074

---

ユティリティコマンドが起動中のため DB ユティリティは実行できません。

### 要因

ユティリティコマンドが起動中のため、DB ユティリティは実行できませんでした。

### 対処

ユティリティコマンド終了後、DB ユティリティを実行してください。または、ユティリティコマンドで作業を実行してください。

## 2077

---

設定項目エラー File:XX Line:YY

### 要因

テキストデータベースのカスタマイズ情報定義ファイル記述で誤りがあった設定項目内容は無効とします。

XX：テキストデータベースのカスタマイズ情報定義ファイルのファイル名

YY：誤りのあった行番号

### 対処

設定項目の項目名と属性を見直し、テキストデータベースのカスタマイズ情報定義ファイルを修正後、ユティリティコマンド TS2DBInfo を再実行してください。

## 2078

---

設定項目エラー File:XX Item:YY

### 要因

テキストデータベースのカスタマイズ情報定義ファイル記述で誤りがあった設定項目内容は無効とします。

XX：テキストデータベースのカスタマイズ情報定義ファイルのファイル名

YY：誤りのあった項目名

**対処**

設定項目の項目名と属性を見直し、テキストデータベースのカスタマイズ情報定義ファイルを修正後、ユティリティコマンド TS2DBInfo を再実行してください。

---

## 2079

テキスト DB(XX) はインクリメンタル処理実行中です。

**要因**

インクリメンタル処理中のテキストデータベースに対して、インクリメンタルを実行しようとした。

XX：テキストデータベース名

**対処**

実行中のインクリメンタル処理が終了してから、再実行してください。

---

## 2080

テキスト DB(XX) は他の処理で使用しています。

**要因**

指定したテキストデータベースはほかの処理を実行中のため、使用できません。

XX：テキストデータベース名

**対処**

実行中の処理が終了してから、再実行してください。

---

## 2081

セッション番号の指定に誤りがあります。

**要因**

指定したセッションは存在しません。

**対処**

セッション番号の指定を見直して再実行してください。

---

## 2082

ソート対象件数が上限件数を超えています。

**要因**

ソート対象件数が上限件数を超えているためソートしません。

**対処**

ソート対象件数を減らした検索結果で再実行するか、または上限件数の指定値を変

更して再実行してください。

## 2083

---

クライアントのバージョン番号が不正です。

**要因**

クライアントのバージョンが古い場合、指定したコマンドは実行できません。

**対処**

指定したコマンドが実行できるクライアントにバージョンアップしてから、再度コマンドを実行してください。

## 2084

---

サーチコマンドの文字コード変換に失敗しました。(XX)

**要因**

サーチコマンドの文字コード変換に失敗しました。

XX：変換に失敗したサーチコマンドのオフセット位置

**対処**

サーチコマンドを見直して、再度実行してください。

## 2085

---

文字コード種別が不正です。

**要因**

使用しているクライアントの文字コード種別、またはテキストデータベースの文字コード種別では、このコマンドは実行できません。

**対処**

クライアントの文字コード種別とテキストデータベースの文字コード種別を一致させてから、再度実行してください。

## 2086

---

登録文書とテキストデータベースの文字コード種別の組み合わせが不正です。(XX)

**要因**

登録文書とテキストデータベースの文字コード種別の組み合わせが不正なため、このコマンドを実行できません。

XX：登録文書とテキストデータベースの文字コード種別

**対処**

登録文書とテキストデータベースの文字コード種別を見直して、再度実行してください。

**2087**

---

全文検索用データ変換定義ファイルの指定に誤りがあります。(XX)

**要因**

全文検索用データ変換定義ファイルの指定に誤りがあるため、XML の変換処理を実行できません。

XX：テキストデータベースと登録文書の文字コード種別

**対処**

全文検索用データ変換定義ファイルの指定を見直して、再度実行してください。

**2089**

---

メモリ常駐化機能が有効のため、MINCREMENTAL は実行できません。

**要因**

メモリ常駐化機能が有効の場合、MINCREMENTAL (テキストデータベースのインクリメンタル) はできません。

**対処**

メモリ常駐化機能を無効にして再実行するか、ユティリティコマンド TS2DBincr でインクリメンタルしてください。

**2090**

---

同義語展開数および異表記展開数の合計が最大展開数 (XX) を超えました。

**要因**

検索条件式に従った検索タームの同義語展開数、および異表記展開数の合計が、サーバー設定ファイルの expand\_counts 項目で指定した最大展開数を超えるため検索を中断しました。

XX：expand\_counts で指定した最大展開数

**対処**

検索タームを変更してから、または同義語展開、異表記展開指定を変更してから再実行してください。

**2091**

---

XX のインデックスの上限値に対する使用率が上限値を超えました。(YY, ZZ)

**要因**

特定のインデックスのインデックスサイズが上限値 (2,147,483,647 バイト) を超えるため登録できません。

XX：文字コード

YY：データベース名

ZZ：検索対象ファイル名

**対処**

不要な登録済み文書を削除して、コンデンスを実行してください。

## 2527

---

指定された物理格納ディレクトリ数が一致しません。

### 要因

コンデンス用物理格納ディレクトリの指定がすでに定義されている物理格納ディレクトリ数と一致しないため、処理が実行できませんでした。

### 対処

テキストデータベースのディレクトリ構成を見直して、物理格納ディレクトリ数が一致するようにコンデンスの定義を変更してください。

## 2528

---

テキスト DB (XX) に対して、本コマンドは実行できません。

### 要因

Web Search 用または Groupmax Document Manager 用のテキストデータベースに文書の登録コマンド、または削除コマンドを実行しようとした。

XX：指定されたテキストデータベース名

### 対処

指定したテキストデータベース名を見直してください。

Web Search 用または Groupmax Document Manager 用のテキストデータベースに対して実行したい場合は、Web Search または Groupmax Document Manager から操作してください。

## 2529

---

検索対象ファイルと関連情報ファイルの名称が重複しています。

### 要因

同一テキストデータベース内で検索対象ファイルと関連情報ファイルの名称が重複しています。

### 対処

ファイル名が重複しないように指定を見直して、再実行してください。

## 2530

---

XX の実行がキャンセルされました。

### 要因

実行中の処理がキャンセルされました。

XX：実行中の処理（文書登録，文書登録前処理，コンデンス，またはインクリメンタル）

**2531**

---

関連情報ファイルの作成できる最大を超えているため、関連情報ファイル (XX) の追加ができません。

**要因**

関連情報ファイルを作成しようとしたが、関連情報ファイルの追加できる数の最大値を超えていたため、追加できませんでした。

XX: 関連情報ファイル名

**2532**

---

DB ユティリティが起動中のためユティリティコマンドは実行できません。

**要因**

DB ユティリティが起動中のため、ユティリティコマンドは実行できませんでした。

**対処**

DB ユティリティが終了してから、ユティリティコマンドを実行してください。または、DB ユティリティで作業を実行してください。

**2533**

---

文書 ID 指定の文書登録前処理が行われているため、実行できません。

**要因**

サーチコマンド MDATAMAKE を実行しようとしたが、文書 ID 指定の文書登録前処理中のため、実行できませんでした。

**対処**

文書登録前処理済みの文書の登録または削除を実行してから、再実行してください。

**2534**

---

文書登録前処理でエラーが発生しました。(XX)

**要因**

サーチコマンド MDATAMAKE 処理中の文書登録前処理でエラーが発生しました。

XX: エラー内容

**対処**

エラー内容を参照してエラーとなった項目を修正し、再実行してください。

**2540**

---

テキスト DB(XX) の関連情報ファイル (YY) には既に範囲指定検索ファイルが作成されています。

**要因**

指定された関連情報ファイルにはすでに範囲指定検索ファイルが作成されています。

XX: テキストデータベース名

YY：関連情報ファイル名

対処

テキストデータベース名または関連情報ファイル名を見直して、再実行してください。

---

## 2541

テキスト DB(XX) の関連情報ファイル (YY) には範囲指定検索ファイルが作成されていません。

要因

指定された関連情報ファイルには範囲指定検索ファイルが作成されていません。

XX：テキストデータベース名

YY：関連情報ファイル名

対処

テキストデータベース名または関連情報ファイル名を見直して、再実行してください。

---

## 2542

テキスト DB(XX) の状態が異常です。(YY)

要因

テキストデータベースの状態が異常です。

XX：テキストデータベース名

YY：異常な DB ディレクトリパス名

対処

テキストデータベース作成時に指定した DB ディレクトリパスを再確認してください。DB ディレクトリが存在しない場合はテキストデータベースを削除してから、再度作成してください。

---

## 2544

登録文書の最上位構造名が登録済みの最上位構造名と異なります。

要因

登録文書の最上位構造名が登録済みの最上位構造名と異なるため、登録できません。

XX：登録文書情報

対処

登録文書の内容、または登録先のテキストデータベース名を確認して、再度登録してください。

---

## 9000

MEMORY SHORTAGE(XX,YY)

要因



処理に必要なメモリをシステムが取得できませんでした。

XX：メモリを要求したソースファイル名

YY：エラーが発生した行

対処

サーバーのメモリの状態を確認してください。

## 9001

---

Invalid Setup Parameter File:XX Line:YY

要因

サーバー設定ファイルの記述に誤りがありました。誤りのあった項目は無効になります。

XX：サーバー設定ファイルのファイル名

YY：誤りのあった行番号

対処

サーバー設定ファイルの項目名と属性を見直して修正してから、サーバーを再起動してください。

## 9002

---

No Define Service Name

要因

services ファイルにサービス名が登録されていません。

対処

services ファイルにサービス名を登録して、再実行してください。

## 9003

---

Invalid Setup Parameter File:XX Item:YY

要因

サーバー設定ファイルの記述に誤りがありました。誤りのあった項目は無効になります。

XX：サーバー設定ファイルのファイル名

YY：誤りのあった項目名

対処

サーバー設定ファイルの項目名と属性を見直して修正後、サーバーを再起動してください。

## 9900

---

FILE IO ERROR(VV,WW,XX,YY,ZZ)

要因

ファイルのアクセス時にエラーを検出しました。

VV：ファイル名

WW：オペレーション

XX：詳細情報

YY：オペレーションを発行したソースファイル

ZZ：オペレーションを発行した行

対処

保守員に連絡してください。

## 9901

---

SYSTEM FILE IO ERROR(VV,WW,XX,YY,ZZ)

要因

システムファイルのアクセス時にエラーを検出しました。

VV：ファイル名

WW：オペレーション

XX：詳細情報

YY：オペレーションを発行したソースファイル

ZZ：オペレーションを発行した行

対処

保守員に連絡してください。

## 9902

---

SYSTEM ERROR(XX,YY,ZZ)

要因

サーバー内部で自己矛盾を検出しました。

XX：詳細情報

YY：エラーの発生したファイル

ZZ：エラーの発生した行

対処

保守員に連絡してください。

詳細情報の内容が "Invalid P/L status." の場合、ユーティリティコマンド TS2DBadj  
を実行してください。

## 9903

---

テキスト DB(XX) 更新中の異常終了を検出しました。

要因

テキストデータベースに異常が発生しています。

XX：テキストデータベース名

対処

テキストデータベースのコンデンスを実行してください。

---

## 付録 G 用語解説

### (英字)

---

#### DTD

SGML で文書を作成するときの、文書構造の定義です。

#### TS サーバー

Bibliotheca2 TextSearch Server Version 2 のことです。

#### ESIS-B (全文検索用データ)

用語解説「全文検索用データ」を参照してください。

### (ア)

---

#### アクション単位ソケット

通信ソケットの使用方法の一つです。セッション確立、セッション切断、サーチコマンド実行のアクション単位ごとにソケットの生成とクローズをくり返します。

#### 重み付け

検索条件を複数指定するときに、各検索条件に対して指定する重要度のことです。

### (カ)

---

#### 簡易 XML 文書解析機能

データベース種別が XML の場合、XML 文書登録時に使用できる機能です。簡易 XML 文書解析機能は、関数、およびコマンドを指定して使用します。この機能を使用すると、外部の XML パーサを使用しないで TS サーバーに文書を登録できます。

#### 概念検索

検索条件として指定した文章（種文章）をキーに、似た内容（概念）を持つ文書を検索する検索方法です。

#### 関連情報ファイル

文書の関連情報（見出し情報や図データなど）を管理するファイルです。

#### 近傍条件検索

単語と単語の間の文字数を条件として指定する検索方法です。

#### 検索結果集合

検索条件に合致した文書の集合です。

### 検索条件

テキストデータベース内の文書を検索するための条件です。

### 検索条件式

検索条件の単位になる式です。サーチコマンド MFIND に指定します。

### 検索セッション

サーチコマンド MSTART を実行してからサーチコマンド MEND を実行するまでの、検索できる状態のことです。

### 検索ターム

検索条件に指定するキーワードです。

### 検索対象ファイル

テキストデータベースに登録されている文書とその関連情報を管理するファイルです。文書を検索するときは、このファイル名を指定して、検索対象のテキストデータベースを区別します。

### 検索用特徴ターム

種文章から抽出した特徴タームの中で、その種文章の概念を表すために採用されたタームのことです。種文章の特徴をより明確に示している特徴タームから順に、検索用特徴タームとして採用されます。

### 構造なし文書用テキストデータベース

構造のないプレーンなテキストデータを管理するためのテキストデータベースです。

### 構造文書用テキストデータベース

SGML で作成された、構造を持つ文書を管理するためのテキストデータベースです。

## (サ)

---

### サーチコマンド

TS サーバーに検索の指示を与えるコマンドです。関数に組み込んで使用します。

### サーバー設定ファイル

TS サーバーの動作環境をカスタマイズするためのファイルです。

### 除外文字検索

検索タームの一部に特定の文字がある場合を除外して、該当する文書を検索することです。

### 全文検索用データ (ESIS-B)

XML 文書の内容を解析して、XML 文書の構造情報とコンテンツを抽出したデータです。

### スコア

検索条件に対して、どの程度の適応度があるかを示した得点です。

### セッション単位ソケット

通信ソケットの使用方法の一つです。セッション確立でソケットを生成し、セッション切断までソケッ

トをクローズしません。

## (タ)

---

### 種文章

概念検索の検索条件に指定する文章のことです。

### テキストデータベース

TS サーバーで管理するデータベースです。文書はテキスト形式で格納されます。

### テキストデータベースのオープン

テキストデータベースを検索できる状態にすることです。

### テキストデータベースのクローズ

テキストデータベースを検索できる状態から解放することです。

### 同義語・異表記展開

検索タームの同義語を同義語辞書から呼び出すことです。異表記は、システムで決められた規則に従って呼び出されます。

### 同義語辞書

同義語の展開方法を定義したファイルです。

### 特徴ターム

概念検索用に種文章から抽出される、種文章を特徴づける単語のことです。

## (ハ)

---

### ハイアラーキモード

直前に作成された検索結果の集合を検索の対象にするモードです。

### 文書

テキストデータベースに登録するテキスト形式のファイルです。

### 文書 ID

文書ごとに付けられる固有の番号です。

### ベース

検索の対象になる文書の集合です。

## (マ)

---

### メモリ常駐化定義ファイル

メモリ常駐化機能を使用する場合に指定するファイルです。TS2DBopen コマンド、または

TS2DBopenmode コマンドで指定できます。

## (ヤ)

---

### ユニバースモード

検索の対象になる文書の集合を固定して検索するモードです。

## (ラ)

---

### ランキング

検索結果集合内の文書を、スコアを基に高得点順にソートすることです。

## (ワ)

---

### ワイルドカード

文字の代わりに指定する記号です。検索する単語の一部しかわからないとき、わかっている部分にワイルドカードを付けて指定します。





---

# 索引

---

## 記号

¥ (円記号) の取り扱い 151

---

## A

AND 条件 14

---

## B

Bibliotheca2 TextSearch Development Kit  
Version 2 3  
Bibliotheca2 TextSearch Extension Version  
2 2  
Bibliotheca2 TextSearch Gateway Version 2  
3  
Bibliotheca2 TextSearch Runtime Library  
Version 2 3  
Bibliotheca2 TextSearch Server Library  
Version 2 3  
Bibliotheca2 TextSearch Server Version 2 2  
Bibliotheca2 TS での検索 12  
Bibliotheca2 TS での検索の概要 13  
Bibliotheca2 TS の概要 2  
Bibliotheca2 TS の特長 6  
Bibliotheca2 TS の目的 2  
Bibliotheca2 TS を構成するプログラム 2  
BOOLEAN 方式 112

---

## C

Cosminexus Version 7 以降を使用する場合  
156

---

## D

Development Kit または Server Library で  
使用する関数 36  
DTD [用語解説] 200

---

## E

ESIS-B (全文検索用データ) [用語解説]  
200

---

## J

JISX0221 と MS-Unicode の文字別全半角対  
応 UTF-8 文字コード 149

---

## L

Library 9  
Library の概要 9  
Library の環境設定 136

---

## M

MCONCEPT 80  
MCONCEPT (概念検索) 82  
MDATAMAKE 80  
MDATAMAKE (文書の追加) 90  
MDATARM 80  
MDATARM (文書の削除) 94  
MDBINF 80  
MDBINF (テキストデータベース情報の出  
力) 95  
MDISPLAY 80  
MDISPLAY (検索結果の文書の出力) 102  
MEND 80  
MEND (テキストデータベースの使用終了  
宣言) 106  
MFIND (全文検索) 80, 107  
MFIND (範囲指定検索) 80, 116  
MINCREMENTAL 80  
MINCREMENTAL (テキストデータベース  
のインクリメンタル (最適化)) 120  
MLOOK 80  
MLOOK (検索タームの同義語・異表記展開  
の出力) 121  
MOPT 80  
MOPT (検索オプションの指定) 124

MOSTAT 80  
 MOSTAT( 検索オプションの設定値の取得 )  
 126  
 MSORT 80  
 MSORT ( 検索結果のソート ) 128  
 MSTART 80  
 MSTART ( テキストデータベースの使用開  
 始宣言 ) 132  
 MSWEEP 80  
 MSWEEP ( 検索結果集合の削除 ) 133

## O

---

OR 条件 14  
 OS が UNIX の場合の Bibliotheca2 TS シス  
 テムの構成例 5  
 OS が UNIX の場合の環境設定 136  
 OS が Windows の場合の Bibliotheca2 TS シ  
 ステムの構成例 4  
 OS が Windows の場合の環境設定 136

## S

---

SGML や XML で書かれた構造を持つ文書の  
 管理と検索 7

## T

---

Text Search Dictionary 3  
 TS2F\_cancelcmd 36  
 TS2F\_cancelcmd ( サーチコマンドの中断 )  
 45  
 TS2F\_canceldat 36  
 TS2F\_canceldat ( サーチコマンドの実行結  
 果の破棄 ) 47  
 TS2F\_cmd 36  
 TS2F\_cmd ( サーチコマンドの実行 ) 49  
 TS2F\_cmd\_DATA 36  
 TS2F\_cmd\_DATA ( 入力データ付きサーチコ  
 マンドの実行 ) 52  
 TS2F\_cmd関数でサーチコマンドを処理する  
 39  
 TS2F\_con 36  
 TS2F\_con ( 通信セッションの確立 ) 55  
 TS2F\_discon 36

TS2F\_discon ( 通信セッションの切断 ) 56  
 TS2F\_geterror 36  
 TS2F\_geterror ( 詳細エラー情報の取得 ) 57  
 TS2F\_gettermpos 36  
 TS2F\_gettermpos ( 検索タームのヒット位置  
 の取得 ) 59  
 TS2F\_mlookcnv 36  
 TS2F\_mlookcnv ( サーチコマンド MLOOK  
 の実行結果の変換 ) 61  
 TS2F\_rcvdat 36  
 TS2F\_rcvdat ( サーチコマンドの実行結果の  
 取得および終了の確認 ) 63  
 TS2F\_sesopt 36  
 TS2F\_sesopt ( セッションオプションの操作 )  
 68  
 TS2F\_sndcmd 36  
 TS2F\_sndcmd ( サーチコマンドの実行の要  
 求 ) 70  
 TS2F\_sndcmd\_DATA 36  
 TS2F\_sndcmd\_DATA ( 入力データ付きサー  
 チコマンドの実行の要求 ) 72  
 TS2F\_sndcmd 関数および TS2F\_rcvdat 関  
 数でサーチコマンドを処理する 40  
 TS2F\_well\_formed\_xmlcpl 36  
 TS2F\_well\_formed\_xmlcpl ( 整形 XML 文  
 書の全文検索データ変換 ) 74  
 TS2F\_well\_formed\_xmlcpl\_free 36  
 TS2F\_well\_formed\_xmlcpl\_free ( 整形 XML  
 文書の全文検索データ格納領域解放 )  
 78  
 TS2SOCKET 137  
 TS2XMLcnv コマンドが出力するエラーメッ  
 セージ 158  
 TSLANG の指定値 137  
 TS サーバー [ 用語解説 ] 200  
 TS サーバーのエラーコード 174

## U

---

UAP を作成する上での注意 42  
 UTF-8 コードで使用できる文字 147  
 UTF-8 コードの JISX0221 と MS-Uncode  
 の違い 148

---

**V**

Visual C++ 42

---

**W**

Windows 上で TS2XMLenv コマンドを実行する場合に設定する環境変数 156

---

**X**

XML ファイルを全文検索用データファイルに変換する 152

XML 文書の制限 153

---

**あ**

アクション単位ソケット 137

アクション単位ソケット〔用語解説〕200

アルファベット異表記展開 15

---

**い**

異表記展開検索〔概念検索〕26

いろいろな検索と条件式の例題 28

インストールパス〔UNIX〕137

インストールパス〔Windows〕136

---

**え**

エスケープ文字への影響 151

エラーコードの形式 174

エラーコードの詳細 174

エラーステータスの形式 162

エラーステータスの詳細 163

エラーステータスの範囲と意味 162

エラーステータスの分類 162

エラーメッセージの形式 158

エラーメッセージの詳細 159

---

**お**

重み（重要度）を付けた検索 23

重み付け〔用語解説〕200

「重み付け」を利用した検索 7

---

**か**

概念検索 7

概念検索〔用語解説〕200

かたかな異表記展開 15

簡易 XML 文書解析機能〔用語解説〕200

関数のエラーステータス 162

関数の概要 36

関数の発行順序 39

関数の文法 35

関連情報 6

関連情報ファイル〔用語解説〕200

---

**き**

近傍条件検索 15

近傍条件検索〔用語解説〕200

---

**く**

繰り返し構造に対する検索 17

---

**け**

検索および登録に使用できる文字 147

検索結果集合 12

検索結果集合〔用語解説〕200

検索結果集合 ID 12

検索結果のソート 24

検索条件〔用語解説〕201

検索条件式 12

検索条件式〔用語解説〕201

検索セッション 81

検索セッション〔用語解説〕201

検索ターム 6, 12

検索ターム〔用語解説〕201

検索ターム間の文字数を条件にした検索 15

検索タームに指定できるワイルドカードと特殊文字 14

検索タームによる絞り込み検索 27

検索タームを 1 語指定した検索 13

検索タームを指定した検索 13

検索タームを複数個指定した検索 14

検索対象の文書 16

検索対象ファイル〔用語解説〕201

検索の概要 11  
検索の条件の指定 13  
検索範囲を条件にした検索 6  
検索用特徴ターム 24  
検索用特徴ターム〔用語解説〕 201  
検索用特徴ターム出力 26

## こ

---

構造なし文書用テキストデータベース〔用語解説〕 201  
構造文書用テキストデータベース 17  
構造文書用テキストデータベース〔用語解説〕 201  
構造名を指定した検索 17  
構造名を指定した検索〔概念検索〕 26  
コマンドの格納先 155

## さ

---

サーチコマンド 9  
サーチコマンド〔用語解説〕 201  
サーチコマンドの一覧 80  
サーチコマンドの概要 80  
サーチコマンドの使用上の注意 81  
サーチコマンドの文法 79  
サーバー設定ファイル〔用語解説〕 201  
作成する UAP の概要 9

## し

---

シフト JIS コードで使用できる文字 147  
種文章〔用語解説〕 202  
種文章を複数指定する検索 26  
条件式 (MCONCEPT) の例題 33  
条件式 (MFIND) の例題 29  
条件式 (MSORT) の例題 33  
使用するファイル 37  
使用するファイル (UNIX の場合) 37  
使用するファイル (Windows の場合) 37  
使用できない文字に関する注意 148  
除外文字検索〔用語解説〕 201

## す

---

スコア 19  
スコア〔用語解説〕 201  
スコア検索 26  
スコア総和方式 113

## せ

---

セッション単位ソケット〔用語解説〕 201  
セッション単位ソケット 138  
全角・半角異表記展開 15  
全半角文字統一機能および全角・半角異表記展開への影響 148  
全文検索 6  
全文検索用データ (ESIS-B)〔用語解説〕 201  
全文検索用データ変換定義ファイルの記述形式 154

## そ

---

ソート (ランキング) 7

## た

---

大量の文書の管理 6  
タグ属性に対する検索 18  
種文章 24

## つ

---

通信環境の設定 136  
通信ソケットの環境設定 137  
通信ソケットの設定 138

## て

---

テキストデータ以外の関連情報の管理 6  
テキストデータベース 2,6  
テキストデータベース〔用語解説〕 202  
テキストデータベースのオープン〔用語解説〕 202  
テキストデータベースのクローズ〔用語解説〕 202  
テキストデータベースの検索方法 12

## と

同義語・異表記展開〔用語解説〕 202  
 同義語・異表記の展開例 16  
 同義語・異表記展開検索 15  
 同義語辞書〔用語解説〕 202  
 同義語展開 15  
 同義語展開検索〔概念検索〕 26  
 同義語や異表記を利用した検索 7  
 登録文字コードと検索ターム文字コードの組み合わせによるヒット有無 149  
 特殊文字 13  
 特徴ターム 24  
 特徴ターム〔用語解説〕 202  
 特定構造検索条件式 18  
 特定の構造に対する検索 17  
 得点（スコア） 7

## に

任意の文字列や文章を手がかりとした検索 7

## は

ハイラーキサーチ 16, 125  
 ハイラーキモード〔用語解説〕 202  
 範囲指定検索 6  
 範囲を指定した検索 15  
 半角アルファベット異表記展開 15

## ふ

複合条件式 17  
 複数の構造を指定した検索 17  
 文書〔用語解説〕 202  
 文書 ID〔用語解説〕 202  
 文章の概念を手がかりとした検索 24

## へ

ベース 16  
 ベース〔用語解説〕 202

## ほ

ポート番号の設定 136

## め

メモリ常駐化機能 7  
 メモリ常駐化定義ファイル 120  
 メモリ常駐化定義ファイル〔用語解説〕 202  
 メモリ使用量 155

## も

文字コード種別の環境設定 137

## ゆ

ユーザー指定のキーワードによる全文検索 6  
 ユーザープログラムの作成例 139  
 ユニバースサーチ 16, 125  
 ユニバースモード〔用語解説〕 203

## よ

用語解説 200

## ら

ランキング〔用語解説〕 203  
 ランキング検索 19  
 ランキング検索でのスコアの算出方法（近傍条件検索の場合） 21  
 ランキング検索でのスコアの算出方法（検索条件式同士の論理演算検索の場合） 23  
 ランキング検索でのスコアの算出方法（検索ターム同士の論理演算検索の場合） 20  
 ランキング検索でのスコアの算出 19

## ろ

論理演算検索 16  
 論理演算検索〔概念検索〕 26  
 論理積 14  
 論理和 14

## わ

ワイルドカード 13  
 ワイルドカード〔用語解説〕 203



# ソフトウェアマニュアルのサービス ご案内

## 1. マニュアル情報ホームページ

ソフトウェアマニュアルの情報をインターネットで公開しています。

URL <http://www.hitachi.co.jp/soft/manual/>

ホームページのメニューは次のとおりです。

マニュアル一覧	日立コンピュータ製品マニュアルを製品カテゴリ、マニュアル名称、資料番号のいずれかから検索できます。
CD-ROMマニュアル	日立ソフトウェアマニュアルと製品群別CD-ROMマニュアルの仕様について記載しています。
マニュアルのご購入	マニュアルご購入時のお申し込み方法を記載しています。
オンラインマニュアル	一部製品のマニュアルをインターネットで公開しています。
サポートサービス	ソフトウェアサポートサービスお客様向けページでのマニュアル公開サービスを記載しています。
ご意見・お問い合わせ	マニュアルに関するご意見、ご要望をお寄せください。

## 2. インターネットでのマニュアル公開

2種類のマニュアル公開サービスを実施しています。

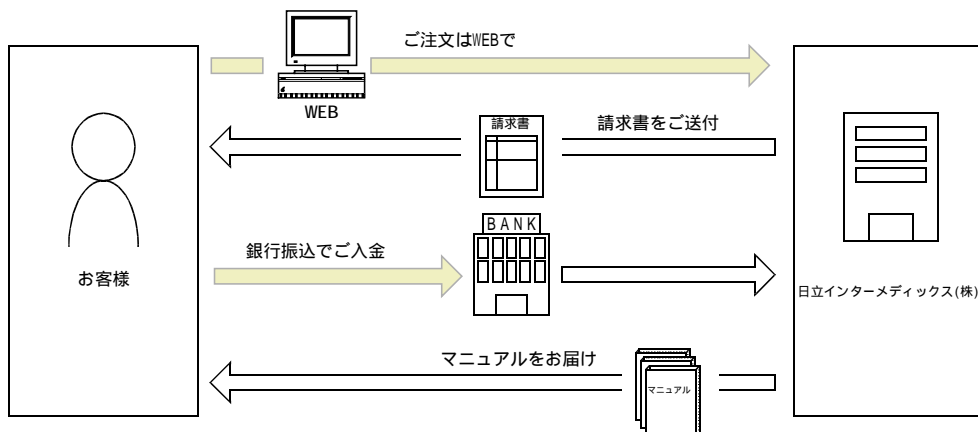
### (1) マニュアル情報ホームページ「オンラインマニュアル」での公開

製品をよりご理解いただくためのご参考として、一部製品のマニュアルを公開しています。

### (2) ソフトウェアサポートサービスお客様向けページでのマニュアル公開

ソフトウェアサポートサービスご契約のお客様向けにマニュアルを公開しています。公開しているマニュアルの一覧、本サービスの対象となる契約の種別などはマニュアル情報ホームページの「サポートサービス」をご参照ください。

## 3. マニュアルのご注文



マニュアル情報ホームページの「マニュアルのご購入」にアクセスし、お申し込み方法をご確認のうえWEBからご注文ください。ご注文先は日立インターメディアックス(株)となります。

ご注文いただいたマニュアルについて請求書をお送りします。

請求書の金額を指定銀行へ振り込んでください。

入金確認後7日以内にお届けします。在庫切れの場合は、納期を別途ご案内いたします。