

# misima 旧仮名遣い・旧字変換支援 利用の手引き

misima Ver.2.3i

安田 功, isao@yasuda.homeip.net

2007年 10月 7日

表記法は音にではなく、語に随ふべし。

—福田恆存

人間の精神には、今ここに流通する言葉から抜け出したい  
という欲求がある。

—水村美苗

stat rosa pristina nomine, nomina nuda tenemus.

過ギニシ薔薇ハ名前ノミ、虚シキソノ名ガ今ニ残レリ。

—Bernard de Morlaix

(U. エーコ『薔薇の名前』より)

## 目次

1	概要	3	2.5	繰返し符号変換機能	10
1.1	開発の背景／目的	4	2.6	仮名反転変換機能	10
1.2	従来システムの問題点	4	2.7	TeX 変換機能	11
1.3	misima の特徴	5	2.8	マーキング機能	12
1.4	利用形態	6	2.9	無変換指定機能	12
1.5	パッケージ内容概略	7	2.10	解析結果トレース機能	13
1.6	動作環境	7	3	インストール	14
2	misima 機能	9	3.1	本体, CGI モジュール	14
2.1	旧仮名遣い変換機能	9	3.2	Emacs Lisp (Elisp)	17
2.2	旧字変換機能	9	3.3	SOAP Webservice	18
2.3	用字・用語変換機能	10	4	misima コマンド	24
2.4	単純変換機能	10	4.1	コマンドライン書式	24
			4.2	.misimarc リソースファイル	25
			4.3	mlatex ユーティリティ	25

5	Emacs からの変換	26	11.1	辞書変換処理	58
5.1	.emacs の設定	26	11.2	本則変換処理	60
5.2	Emacs コマンド	26	11.3	品詞依存処理	60
6	misima Web 環境	31	11.4	拗音・促音変換処理	60
6.1	フォームのカスタマイズ	31	12	用字・用語変換仕様	60
6.2	misima.conf	31	13	単純変換仕様	60
6.3	misima Web 変換フォームの使い方	31	14	繰返し符号変換仕様	60
7	SOAP Webservice	35	15	TEX 変換仕様	61
7.1	Java クライアント環境	35	15.1	多言語文字変換	61
7.2	SOAP Java クライアントのインストール	36	15.2	漢文訓点変換	76
7.3	SOAP Java クライアントの実行	40	15.3	くノ字点変換	78
7.4	Microsoft Word SOAP クライアント	42	16	変換精度の制約・限界	79
8	入力テキスト仕様	49	16.1	辞書要因	80
9	misima 辞書・TEX 変換テーブル	49	16.2	字音仮名遣いの問題	80
9.1	旧字変換定義	49	16.3	形態素解析要因	80
9.2	旧仮名変換定義	51	16.4	文脈要因	81
9.3	用字・用語変換定義	52	16.5	入力表記要因	81
9.4	単純変換定義	53	17	謝辞	81
9.5	TEX 文字変換定義	54	18	利用条件	82
9.6	TEX 異体字変換定義	55	19	更新履歴	83
9.7	misima 辞書登録時の注意事項	56	付録 A	システム辞書：新字-旧字変換対応表	
10	処理方式概念	58		(訂 5)	84
10.1	日本語形態素解析	58	参考文献		90
10.2	変換処理シーケンス	58			
11	旧仮名変換仕様	58			

- 本稿はインストールについては対象読者として、ある程度の OS/Web システム管理知識を有する UNIX ユーザを想定している。
- 本稿で言及しているハードウェア、ソフトウェアの名称はいずれも開発元、販売元の商標もしくは登録商標である。本稿では ™, ® を明示していない。
- 本文書で述べている事項については、実機確認の上、注意を払って記述しているが、誤りが含まれる可能性がある。ご指摘いただくと幸いである。運用の結果については無保証であることを予めご了承ください。
- コマンド、ファイル名、パラメータ、変数、URL は “Typewriter” のように表記している。

## 1 概要

misima 旧仮名遣い・旧字変換支援 Ver.2.3 は、現代仮名遣い・常用／当用漢字<sup>\*1</sup>による UTF-8 テキストを歴史的仮名遣い<sup>\*2</sup>・旧字体／正字体／異体字<sup>\*3</sup>テキストに変換するプログラムである。

変換例を図 1 (3 頁) に、出力を pTeX で組版した結果を図 2 (3 頁) に示す。これは旧仮名遣い変換、旧字変換、用字・用語変換、TeX 変換指定を行ったものである。

```
margarita:/Users/isao[1001] % cat test.txt
森鷗外は明治四十一年、政府が招集した仮名遣い臨時調査委員会で、当時からあった発音に基づく仮名遣い制定問題について意見を述べており、私はその議事録を岩波鷗外選集第十三巻（昭和五十四年刊、全二十一巻）で読んで非常に興味深いものがあった。
鷗外は、ことばが変化するという本性から表記は必然的に歴史的なものとならざるをえず、従って表記は書かれたものに準拠すべきであって音声とは分けて考えるべきであると述べ、政府が性急に仮名遣いを制定することが言語生活に混乱をもたらすと警鐘を鳴らした。
戦後の仮名遣いの論争をみると、その先見の明に感嘆してしまう。
margarita:/Users/isao[1002] % misima -kyitq -s c -x u <test.txt
森\CID{7646}外は\CID{14052}治四十一年、政府が招集した假名\CID{13754}ひ臨時\CID{13933}査委員会で、当時からあつた發\CID{13664}に基づく假名\CID{13754}ひ制定問題について\CID{13639}見を\CID{13822}べてをり、私はその議事\CID{13402}を岩波\CID{7646}外\CID{13887}集第十三巻（昭和五十四年刊、\CID{13890}二十一巻）で読んで非常に興味深いものがあった。
\CID{7646}外は、ことばが變\CID{13665}するといふ本性から表記は必然\CID{13945}に\CID{13398}\CID{13451}\CID{13945}なものとならざるをえず、従つて表記は書かれたものに準拠すべきであつて\CID{13664}聲とは\CID{13499}けて\CID{13445}へるべきであると\CID{13822}べ、政府が性\CID{13712}に假名\CID{13754}ひを制定することが言語生活に混乱をもたらすと警鐘を鳴らした。
戦後の假名\CID{13754}ひの論争をみると、その先見の\CID{14052}に感\CID{13915}してしまふ。
margarita:/Users/isao[1003] %
```

図 1 変換例

森鷗外は明治四十一年、政府が招集した假名遣ひ臨時調査委員会で、当時からあつた發音に基づく假名遣ひ制定問題について意見を述べてをり、私はその議事録を岩波鷗外選集第十三巻（昭和五十四年刊、全二十一巻）で読んで非常に興味深いものがあった。鷗外は、ことばが變化するといふ本性から表記は必然的に歴史的なものとならざるをえず、従つて表記は書かれたものに準拠すべきであつて音声とは分けて考へるべきであると述べ、政府が性急に假名遣ひを制定することが言語生活に混乱をもたらすと警鐘を鳴らした。戦後の假名遣ひの論争をみると、その先見の明に感歎してしまふ。

図 2 変換結果 pTeX 出力例

本文書は misima 関連ソフトウェアの機能とその仕様、インストール、利用方法について示す。misima パッケージは以下の内容から構成されている。

\*1 以下「新字」と称する場合がある。いわゆる略字も含める。

\*2 以下「旧仮名遣い」あるいは単に「旧仮名」と称する場合がある。

\*3 これらを以下まとめて「旧字体」あるいは単に「旧字」と称する場合がある。

- 変換機能を実行する Perl プログラム本体 `misima`
- 辞書 (旧仮名・旧字変換用システム辞書 /  $\text{T}_{\text{E}}\text{X}$  変換テーブル / 茶筌用ユーザ辞書)
- Web ブラウザからのアクセスを処理する CGI モジュール `misima.cgi`
- GNU Emacs から `misima` を利用するための Emacs Lisp `misima.el`
- SOAP Webservice `misima` サーバ `misimaSoapConnector.class`
- SOAP Webservice Java クライアント `misimaSoapClient.class`
- GNU Emacs から SOAP サーバを利用するための Emacs Lisp `misimasoap.el`
- SOAP Webservice Microsoft Word クライアント `misimaTemplate.dot` (テンプレート)
- SOAP サーバ環境用ファイル (WSDD, `log4misima.properties` 等)
- 辞書管理ツール<sup>\*4</sup>,  $\text{T}_{\text{E}}\text{X}$  コンパイル支援のためのユーティリティ `mlatex`
- README, 本ドキュメント `misimadoc.pdf`

`misima-2.3` パッケージアーカイブの最新版は <http://yasuda.homeip.net/dl/dl.html> からダウンロードできる。

## 1.1 開発の背景 / 目的

`misima` のもともとの開発の動機は、筆者が中央公論社版『谷崎潤一郎全集』の旧仮名遣い・旧字体の文章から長い引用をする必要に迫られたことに他ならない。国文学の研究論文などで、明治・大正・戦前昭和の文学の長大な引用を行わなければならないとき、歴史的表記の引用文を入力するのは容易ではない。

それは、まず第一に現代仮名遣いに最適化された通常の仮名漢字変換ソフトウェア / インputメソッド (以下 IME) では、旧仮名表記での入力が辞書に適合せず、正しい仮名漢字変換がなされないためである。第二に、漢字についても同様に常用漢字で表記することが現代文では正則であるために、IME で旧字体が変換候補に上がることは希である。これには IME が現時点では JIS 第一・二水準の漢字に変換せざるをえないという状況も関係している。<sup>\*5</sup> さらに戦前の表記は用字・用語において現代表記と異なる用例が多い。また戦前の法律文は旧仮名・旧字であるほかに、もっぱら片仮名主体の文章である。これらも IME 変換に不適な特質のひとつである。

`misima` は主として学術論文においてこのような引用文を効率的に作成することを目的としている。現代仮名遣い、新字表記による現代文に最適化された計算機を用いて、独自の辞書により機械的に旧仮名遣い、旧字体、用字・用語に変換する。つまり、これにより容易に入力できると同時に、変換結果を少ない労力で修正し、目的とする引用文に早く到達できるだろうという考え方である。

## 1.2 従来システムの問題点

同じようなプログラムがいくつか出回っている。ところが、これらは引用という目的をもった計算機変換という趣旨に限れば、問題点がいくつか筆者の目につく。それは次のようなものである。

---

<sup>\*4</sup> `module/` ディレクトリ下には `misima`, `misima.cgi` の主機能モジュール以外に、`gensdiclist` (辞書登録旧字定義文字の一覧を  $\text{T}_{\text{E}}\text{X}$ , HTML 形式で出力する), `chadic` (茶筌辞書と `misima` システム辞書との間でエントリの相互チェックを行う), `includechadic` (茶筌辞書から `misima` 辞書登録候補を抽出する) などが含まれている。しかしながらこれらは筆者の開発ツールであり、本稿においてはとくに説明を行わない。

<sup>\*5</sup> Mac OS X, Windows Vista では Unicode 文字への変換もできるようになっている。

- 国語国字の伝統文化の観点から問題点をあげつらわれている JIS 第一、二水準の文字コードしかサポートしていない。
- 旧仮名変換機能において単語の特性を無視して「え」を「へ」に全面的に置き換えるなど、あまりに機械的なものがある。
- プログラムの目的が擬古文の作成そのものにより、口調をことさら古めかしくするなどの文体的調整を施すものがある。これは引用を基本とした用途には相応しくないと筆者は考える。国語学的にも「一」を「壹」に、「～ということ」を「～と云ふ事」に変換する必要はないはずである。文章に紋付袴を被せることは misima の主目的でない。
- Web や専用 GUI のユーザインタフェースがほとんどであり、バッチでの利用を想定していない。このため、`sort`、`sed`、`grep` などのテキストユーティリティと組合わせた活用ができないだけでなく、大量もしくは多数のファイルの変換において著しく効率が悪い。また他のアプリケーションとの連携が難しく、利用者は Web ブラウザ、当該プログラムが動作する環境でしか利用できず、かつ変換結果をコピー・ペーストで二次利用せざるをえない。
- 学術論文という最終成果物の作成を支援する機能が考慮されていない。

旧仮名遣い・旧字体変換において、完全は無理にせよ上記問題点をカバーし、かつ高い変換精度を実現したいとの問題意識・関心から、単語の文法解析・辞書に基礎をおいた変換ツールを作ろうと一念発起したわけである。とはいえ、計算機で一定の基準に従って一気に変換し、かつ辞書に依存する方式であり、もとよりサラリーマンの日曜大工であってみれば、多様な国語表記を完全無欠に変換するのは無理である。

### 1.3 misima の特徴

misima の設計に際して基本とした事項を以下に示す。これが取りも直さず 1.2 節 (4 頁) で述べた問題点へ解であり、本パッケージの特徴である。

#### 語の文法解析への準拠

語の品詞及び活用形を解析した結果で変換判断を行う。「語」の切出しと文法解析のため、日本語形態素解析ソフトウェア茶筌を利用する。福田恆存の「表記は語に随ふべし」との言に倣ったつもりである。

#### UTF-8 サポート

UTF-8 コードテキストを入出力前提としており、JIS 第一・二水準の範囲を越えた文字を取り扱うことができる。<sup>\*6</sup>

#### Adobe-Japan1 文字空間の活用

旧字変換において、UTF-8 文字出力だけでなく、 $\text{T}_{\text{E}}\text{X}$  OTF パッケージの CID 番号参照形式でも出力できるようにした。OTF パッケージは齋藤修三郎氏の開発した  $\text{T}_{\text{E}}\text{X}$  Virtual Fonts 及びスタイルファイルである。これにより、DTP において利用が進んでいる Adobe-Japan1.5 エンコーディング 2 万字の広範囲な文字空間から、JIS、Unicode では定義されていない文字 (主に異体字) も  $\text{T}_{\text{E}}\text{X}$  文書で使うことが可能となる。OTF を用い、ヒラギノ Pro や小塚 Pro といった Adobe-Japan1.5 対応フォントでタイプセットすると、素晴らしい出力が得られるはずである。

<sup>\*6</sup> misima パッケージ添付の Web のフォームを利用する場合、入力の文字コードを意識する必要はないが、旧字体出力結果は Unicode フォントがないと化けてしまうことが想定される。最近のブラウザならまず Unicode のページを表示できるはずである。ただし、ブラウザによっては「祈」(「祈」の旧字体) など、人名許容もしくは康熙字典別掲漢字の一部が出力されない。

## 多言語 TeX 対応

学術論文という成果物への指向として、筆者は TeX 文書作成を基本とした。旧仮名遣い・旧字の引用を必要とする論文作成者は、一方で外国語を取り扱う文科系研究者であると想定し、misima は主要ヨーロッパ言語、古典ギリシア語、スラヴ語の混在した多国語テキストを pTeX で処理可能な形式に変換する機能を実装した。

## 古典籍引用の考慮

漢文訓点を必要最小限の入力で TeX sfkanbun 形式に変換する機能、和文繰返し表現を自動的に TeX OTF くノ字点コントロールシーケンスに変換する機能をサポートした。TeX 変換機能には、旧仮名遣い・旧字体の和文のみならず、古典・漢文、文科系論文を書くにあたって必要と思われる支援を盛り込んだつもりである。

## フィルタ構造

変換エンジンはフィルタ（標準入出力を扱うコマンド）として動作する構造とした。他のテキスト・ユーティリティと併用したバッチ大量処理ができる。他のプログラムから呼び出すことが容易であるため、Web ブラウザ、GNU Emacs、Microsoft Word からの利用など、利用者の型・目的に応じた応用が可能になった。misima をご利用のエディタやプログラムから呼び出して活用することも容易である。

## 1.4 利用形態

misima は Perl 言語による misima 本体モジュールが機能の中核である。これを呼び出すことによりローカル、リモートでいくつかの利用形態をサポートしている。ローカルはスタンドアローン UNIX 計算機内での利用形態である。リモートはリモートの UNIX 計算機上の misima プログラムをネットワーク越しに操作する形態である。

本パッケージはそれぞれの形態についてユーザインタフェースによって複数の利用方法を提供している。

### コマンドライン（ローカル）

ローカル計算機にある misima コマンドを、仮想端末（ターミナル）シェルのコマンドラインから直接起動する。当然ながら、シェルスクリプト（バッチファイル）からコールして一連のテキスト処理の一部として組み込んで利用することもできる。

### GNU Emacs からの操作（ローカル）

GNU Emacs<sup>\*7</sup>から、ローカル計算機にある misima 本体を呼び出す利用形態である。エディタで文書を作成しつつ必要な領域を指定して misima 変換を実行する。GNU Emacs は UNIX ワールドにおいてデファクトスタンダードといってもよい高機能な多言語テキストエディタである。Windows においても Meadow<sup>\*8</sup>という名で移植されている。

### Web ブラウザ（リモート）

Web ブラウザから CGI フォームによってリモートの misima Web サーバに変換指示を行う形態である。

### SOAP Webservice（リモート）

Webservice クライアントからリモートの misima SOAP サーバに変換指示を行う形態である。添付

<sup>\*7</sup> <http://www.gnu.org/software/emacs/>

<sup>\*8</sup> <http://www.meadowy.org/meadow/>

WebService クライアントにはコマンドラインと Emacs 及び Microsoft Word の三つのユーザインタフェースがある。

## 1.5 パッケージ内容概略

本パッケージ・アーカイブの内容概略を表 1 (7 頁) に示す。

表 1 misima アーカイブ内容

リソース	内容概略
<a href="#">README</a>	<a href="#">README</a> (パッケージ説明)
<a href="#">Makefile</a>	インストール用 <a href="#">Makefile</a>
<a href="#">module/</a>	プログラム本体, CGI モジュール, 管理ツール
<a href="#">dic/</a>	misima 辞書, <a href="#">TeX</a> 変換テーブル
<a href="#">dic/chasen</a>	茶筌用ユーザ辞書
<a href="#">dic/contrib</a>	拡張変換テーブル
<a href="#">env/</a>	環境管理ファイル
<a href="#">html/</a>	Web フォーム (プロトタイプ)
<a href="#">doc/</a>	機能説明ドキュメント, <a href="#">TeX</a> サンプル文書
<a href="#">elisp/</a>	Emacs 用 <a href="#">Elisp</a> (Emacs Lisp)
<a href="#">soap/</a>	SOAP WebService 用リソース
<a href="#">soap/src</a>	SOAP サーバ/クライアント Java ソース
<a href="#">soap/conf</a>	デプロイ用リソース, WSDL
<a href="#">soap/lib</a>	コンパイル済み SOAP クライアント <a href="#">misima.jar</a>
<a href="#">soap/tools</a>	SOAP 用ラッパースクリプト, Word テンプレート, SOAP 用 <a href="#">Elisp</a>

## 1.6 動作環境

misima のインストール, 実行のための動作環境・前提条件を以下に示す。

- Perl 5.8.0 以上 (Unicode サポート版) が必要である。Mac OS X Tiger 10.4, FreeBSD 6.2-RELEASE にて動作確認を行っている。Linux では動作すると考えられる (未確認)。Windows では動作は保証されない。<sup>\*9</sup>
- 日本語形態素解析ソフトウェア茶筌 ChaSen 2.3.3<sup>\*10</sup>, 同辞書 ipadic 2.7.0 以降が事前にインストールされていなければならない。

<sup>\*9</sup> Windows 版茶筌が UNIX 版と同等の機能・インタフェースを備えていれば動作するかも知れない。一方, Windows 環境でも Windows 上の UNIX クローンである Cygwin (<http://www.cygwin.com/>) を導入し, この環境下でインストール・運用ができるかも知れない。ただし, いずれも筆者は未確認である。[TeX](#) タイ語変換は Windows では動作しない。

<sup>\*10</sup> <http://chasen-legacy.sourceforge.jp/>

- ChaSen Perl モジュールが使用可能でなければならない。茶筌のインストールについてはパッケージ添付のドキュメントを参照。
- ipadic 各辞書は UTF-8 エンコーディングで作成されていなければならない。ファイル一式を EUC-JP から UTF-8 にコード変換し、“`makemat -i w`”, “`makeda -i w`” で UTF-8 辞書を作成すればよい。ipadic UTF-8 辞書の作成方法は <http://chasen.naist.jp/hiki/ChaSen/?FAQ> を参照のこと。インストール手順 (3.1.2 節・14 頁) にもコマンド発行の流れを示している。
- Emacs Lisp ツールを利用する場合、Emacs 21.3 以降 (+Mule-UCS) もしくは Emacs 22.0.50 を予めインストールしておく必要がある。Unicode フォントも利用可能にしておく。<sup>\*11</sup>
- ブラウザからの指示で実行できる Web CGI モジュールを利用するためには HTTP 1.1 をサポートした HTTPD が必要である。筆者は Apache 2.2.3 で確認している。
- タイ語  $\text{\TeX}$  変換 (`-x t|T` オプション) を使用する場合、タイ語単語境界マーキングプログラム `swath`<sup>\*12</sup> 及びコード変換ユーティリティ `iconv`<sup>\*13</sup> が予めインストールされていなければならない。`swath` の組込みと活用については拙稿「CJK, Unicode パッケージ, ロシア語多書体の利用: タイ語単語分割<sup>\*14</sup>」を参照。
- SOAP Webservice 関連モジュールのコンパイル及び実行には、JDK 5.0 (バージョン 1.5.0) 以降の Java 開発・実行環境, JSP/Servlet エンジン Jakarta-Tomcat 4.1 以降<sup>\*15</sup>, SOAP フレームワーク Apache-Axis 1.2.1 以降<sup>\*16</sup> 及びその前提クラスライブラリが必要である。筆者は Tomcat 5.5.17, Axis 1.4.1 で確認している。SOAP サーバは `System.getProperties()` メソッドを用いており、古いバージョンの JDK だと問題が発生するかも知れない。JDK 5.0 以降を推奨する。
- 添付 SOAP クライアント `misimaSoapClient` についてはさらに GNU `getopt` Java port ライブラリ<sup>\*17</sup> も予めインストールしておく。
- Java クラスライブラリは `CLASSPATH` に設定されていなければならない。
- Microsoft Word クライアントの利用は、Windows 2000/XP Word 2003 以降, Microsoft Office XP Web Service Toolkit 2.0 が必要である。
- $\text{\TeX}$  変換機能の変換結果を組版するために  $\text{\LaTeX}$  2<sub>ε</sub>, DVIWARE, 関連スタイル/言語パッケージが必要である。機能をカバーする  $\text{\LaTeX}$  パッケージについては 5 節 (12 頁) に一覧を示している。
- 説明にあるインストール方法は GNU `make` を前提としている。

---

<sup>\*11</sup> FreeBSD, Mac OS X, Windows での Emacs Unicode 環境の調整については次に纏めた。「Emacs 21 Unicode 環境」<http://yasuda.homeip.net/rus2/emacs21.html>, 「Windows Meadow 3.00」<http://yasuda.homeip.net/rus2/meadow3.html>, 「Mac OS X Tiger, X11-Emacs 22.0.50」<http://yasuda.homeip.net/rus2/mac-emacs22.html>.

<sup>\*12</sup> <http://www.cs.cmu.edu/~paisarn/software.html>

<sup>\*13</sup> <http://www.gnu.org/software/libiconv/>

<sup>\*14</sup> <http://yasuda.homeip.net/tex/cjk.html>

<sup>\*15</sup> <http://tomcat.apache.org/>

<sup>\*16</sup> <http://ws.apache.org/axis/>

<sup>\*17</sup> <http://www.urbanophile.com/arenn/hacking/download.html>



## 2 misima 機能

本節では misima 機能を示す。各変換仕様詳細については 11 節 (58 頁) 以降を参照。

### 2.1 旧仮名遣い変換機能

新仮名遣い表記入力テキストを行単位に茶筌形態素解析にかけ、解析結果を語単位に旧仮名表記に変換する。旧仮名変換辞書とシステム内の論理処理とで変換を行う。論理処理は本則処理と品詞依存処理からなる。本則処理は文献 [3] で解説している考え方に基づいた手続きである。品詞依存処理は語活用で考慮すべき変換手続きである。

### 2.2 旧字変換機能

新字を旧字に変換する。

本機能は新字-旧字を一对一で定義した旧字変換辞書に依存する。

ひとつの新字を複数の旧字で書き分ける場合がある。「弁」は「弁護」においては「辯」に、「花卉」においては「瓣」に変換すべきである。このように意味によって旧字体を出力し分けたい場合は、単純変換機能または用字・用語変換機能を併用する。

変換オプションによって出力形式を選択できる。選択可能な出力形式を表 2 (9 頁) に示す。

表 2 旧字変換出力形式

出力形態	内容
UTF-8 文字	UTF-8 エンコードで変換結果文字を出力する。
HTML 数値参照	“&#xFFFF;”形式で旧字を出力する。FFFF は十六進数。
TeX OTF UCS 十六進数値参照	“\UTF{FFFF}”形式で旧字を出力する。FFFF は十六進数。
TeX OTF CID 数値参照	“\CID{99999}”形式で旧字を出力する。99999 は十進数。
TeX 今昔文字鏡番号参照	“\MO{999999}”形式で旧字を出力する。999999 は十進数。 “\MO”部分はユーザが定義で変更できる。

今昔文字鏡については、<http://www.mojikyo.org/> を参照。misima 公開パッケージに含まれる旧字変換辞書 (9.1 節・49 頁) は、今昔文字鏡番号エリアが 0 (UTF-8 旧字で代用) に再設定されている。これは今昔文字鏡のライセンス規定により、今昔文字鏡番号の検索用インデックス公開が制限されているからである。今昔文字鏡番号変換機能を利用する場合は、今昔文字鏡 Character Map ソフトウェアを用いて必要な文字の番号を調べ、misima 辞書にユーザ自身で登録する。今昔文字鏡番号は、大修館書店刊行の諸橋轍次編『大漢和辞典』で採られている大漢和番号を踏襲している。TeX 今昔文字鏡番号参照機能の活用においては今昔文字鏡のライセンス規定に準拠しなければならない。

## 2.3 用字・用語変換機能

明治、大正、戦前昭和の文章には、「退廢」を「頽廢」と書くなど、現在とは異なる表記を行う語がある。時代や書き手の属する階層などの状況により整理するのは一筋縄では行かないが、こうした用字・用語を辞書定義に従って変換する機能をサポートしている。

「ドイツ」を「獨逸」と変換するなど「余計なお世話」と思われる方は、機能をオフにすべきである。

## 2.4 単純変換機能

旧仮名遣い変換、用字・用語変換は茶釜の単語分割・品詞解析に依存する。これに対し単純変換は、茶釜の形態素解析結果に因らず、強制的に文字列を置き換える。

マーキング (2.8 節・12 頁) の対象外である。

添付 Web CGI では、旧仮名遣い変換機能または旧字変換機能が指定されていると、単純変換機能もオンに設定されるようになっている。

## 2.5 繰返し符号変換機能

茶釜の単語分割・品詞解析に基づいて、同一語において同じ平仮名、片仮名、漢字が連続するとき二つ目の文字を文字種によってそれぞれ「ゝ、ゞ」「ゝ、ゞ」(以上一ツ点)「々」(同ノ字点)に変換する。

マーキング (2.8 節・12 頁) の対象外である。

「々」文字を「ゞ (ゆすり点)」にしたければ misima 辞書単純変換定義に登録すればよい。

例を表 3 (10 頁) に示す。

表 3 繰返し符号変換例

変換前	変換後
やや	やゝ
ぢぢ	ぢゝ
益益	益々
ただただ	たゞゞ
ばば [婆] はハハハと笑った	ばゝはハゝゝと笑った
パパとママ	パパとママ

## 2.6 仮名反転変換機能

同じ平仮名と片仮名を相互で反転する。通常の平仮名を主とする文章を、谷崎潤一郎の『鍵』や古い法律文のような片仮名主体の表記にする。

「ヴ」「ヰ」「ヱ」「ヱ」はそれぞれ「うぁ」「うい」「うえ」「うお」に変換する。

マーキング (2.8 節・12 頁) の対象外である。

例を表 4 (11 頁) に示す。

表 4 仮名反転変換例

変換前	変換後
これはペンです。	コレハペンデス。
過ぎにし薔薇は名前のみ	過ギニシ薔薇ハ名前ノミ

## 2.7 T<sub>E</sub>X 変換機能

多言語ならびに縦組和文・漢文を組版したい T<sub>E</sub>X ユーザの支援を目的としている。

マーキング (2.8 節・12 頁) の対象外である。

次に示す変換を指定できる。

- 多言語文字変換

多言語文字変換は UTF-8 で記述された多言語文字を pT<sub>E</sub>X で処理可能な 7 ビット ASCII 形式に変換する。テーブルに登録されていない UTF-8 文字はそのまま出力される。<sup>\*18</sup>

- 標準変換 (ラテンアクセント文字, キリル文字, ギリシア文字, 教会スラヴ語, 異体字, Unicode 漢字)
- 中韓変換 (中国語簡体字, 中国語繁体字, 韓国語・ハングル)
- タイ語変換 (単語境界付加・十六進数変換)
- 拡張変換 (ヴェトナム語・グルジア語・機種依存文字)
- キリル T2A 変換
- 十六進数変換
- 強制変換

- 漢文訓点変換

藤田眞作先生 [sfkanbun.sty](#) の漢文訓点フォーマットに整形する。

- くノ字点変換

繰返し表現を齋藤修三郎氏 OTF パッケージでサポートされているくノ字点出力シーケンスに変換する。

T<sub>E</sub>X 変換機能 T<sub>E</sub>X 出力の全範囲を L<sup>A</sup>T<sub>E</sub>X で処理するためには, 使用する機能に応じて表 5 (12 頁) に示す各 L<sup>A</sup>T<sub>E</sub>X 外部パッケージが必要である。CTAN (Comprehensive T<sub>E</sub>X Archive Network) または開発者のサイトより必要なパッケージをダウンロードして T<sub>E</sub>X システムにインストールしておく。「(L<sup>A</sup>T<sub>E</sub>X 標準)」としたものは teT<sub>E</sub>X をインストールすれば標準で組み込まれるものである。<sup>\*19</sup>

<sup>\*18</sup> T<sub>E</sub>X 変換機能の利用については, 拙稿「misima T<sub>E</sub>X 変換の活用」[http://yasuda.homeip.net/misima/misima\\_tex.html](http://yasuda.homeip.net/misima/misima_tex.html) も参照いただきたい。

<sup>\*19</sup> 土村氏による ptetex3, 角藤先生による W32T<sub>E</sub>X でも標準で組み込まれる。

表 5 T<sub>E</sub>X 変換前提 L<sup>A</sup>T<sub>E</sub>X パッケージ

OTF	和文 OpenType Virtual Fonts とスタイル <sup>*1</sup>
Babel	マルチリンガル (L <sup>A</sup> T <sub>E</sub> X 標準)
Cyrillic バンドル	OT2, T2A, T2B, T2C, X2 エンコーディングサポート (L <sup>A</sup> T <sub>E</sub> X 標準)
Unicode	Unicode 入力, Cyrillic T2D エンコーディングサポート <sup>*2</sup>
Teubner	ギリシア古典文献処理 <sup>*3</sup>
ThaiL <sup>A</sup> T <sub>E</sub> X	タイ語 <sup>*4</sup>
VnT <sub>E</sub> X	ヴェトナム語 <sup>*5</sup>
mxedruli	グルジア語 <sup>*6</sup>
sfkanbun	漢文訓点 <sup>*7</sup>
OldSlav	教会スラヴ語 <sup>*8</sup>
Marvosym	記号 <sup>*9</sup>
Textcomp	記号 (L <sup>A</sup> T <sub>E</sub> X 標準)

<sup>\*1</sup> <http://psitau.at.infoseek.co.jp/otf.html>

<sup>\*2</sup> <http://www.tug.org/tex-archive/macros/latex/contrib/unicode/>

<sup>\*3</sup> <http://www.ctan.org/tex-archive/macros/latex/contrib/teubner/>

<sup>\*4</sup> <http://linux.thai.net/plone/TLWG/thailatex/>

<sup>\*5</sup> <http://vntex.sourceforge.net/download/vntex/>

<sup>\*6</sup> <http://www.tug.org/tex-archive/fonts/georgian/mxedruli/>

<sup>\*7</sup> <http://homepage3.nifty.com/xymtex/fujitas2/texlatex/>

<sup>\*8</sup> <http://yasuda.homeip.net/oldslav/oldslavtex.html>

<sup>\*9</sup> <http://www.tug.org/tex-archive/macros/latex/contrib/eurofont/marvosym/>

## 2.8 マーキング機能

オプション指定により、変換部分のマーキングを行う。これは、misima による変換結果をブラウザや T<sub>E</sub>X 組版結果ビューアで容易に確認できることを目的としている。

辞書変換、旧仮名遣い本則変換でそれぞれ個別にマーキングできる。

変換部分をマーキングするための HTML タグまたは T<sub>E</sub>X コントロールシーケンスをユーザが定義できる。

ただし Web CGI は、セキュリティ上の理由から、HTML タグを“&lt;”、“&gt;”に置換して出力するので、出力効果の確認は実際に表示させる HTML にそれぞれ“<”、“>”を埋め込んで利用する必要がある。

## 2.9 無変換指定機能

`<misima_noop>` 制御タグと `</misima_noop>` 制御タグでテキストを囲むと、当該テキストを変換対象としない。この制御タグは出力されない。

ただし、T<sub>E</sub>X 多言語文字強制変換 (15.1.2 節・65 頁) を指定した場合は、タグ範囲内でも多言語文字変換

処理を行う。

HTML 原文に指定しても未サポートのタグとして扱われるため影響はないはずである。TeX 原稿の場合は、本タグをコメント行に埋め込んでおくと、misima 変換をしない場合でも TeX 処理に影響しない。

旧仮名変換等茶筌辞書に依存する変換を文中の一部のみ抑止したい場合、語の切れ目などを考慮してタグを付与しないと、茶筌が誤解析をしてしまう可能性がある。文・節の切れ目の単位で指定することを推奨する。

## 2.10 解析結果トレース機能

茶筌による形態素解析結果、misima 辞書（旧仮名変換定義、用字・用語変換定義）の引当て状況をトレースして出力する。

これはデバッグ用である。茶筌用ユーザ辞書、misima 辞書の追加などのメンテナンスのために用意している。「語分割が不正な場合は茶筌辞書に追加」、「語解析が正しいのに変換結果が不正な場合は misima 辞書に追加」、等の判断を行うためのものである。

入力文、茶筌解析結果、misima 辞書検索結果、活用語の misima 辞書による変換過程、変換結果、オプション指定内容を標準出力に出力する。茶筌解析・misima 辞書検索結果部分の出力例を図 3 (13 頁) に示す。「[旧]:」、「[用]:」はそれぞれ旧仮名遣い変換辞書、用字・用語辞書のヒットしたエントリを示している。茶筌解析結果における文法用語・コード仕様詳細については、茶筌マニュアル（文献 [1, 2]）を参照のこと。

[入力]:

しわを伸ばしたようなこわいろを使って

しわ	しわ (基本形)-2(名詞) [旧]:しわ 0 [用]:(無)
を	を (基本形)-61(助詞) [旧]:(無) [用]:(無)
伸ばし	伸ばす (基本形)-47-12-4(動詞-五段・サ行-連用形) [旧]:(無) [用]:(無)
た	た (基本形)-74-54-1(助動詞-特殊・タ-基本形) [旧]:(無) [用]:(無)
よう	よう (基本形)-23(名詞) [旧]:やう 0 [用]:(無)
な	だ (基本形)-74-55-6(助動詞-特殊・ダ-体言接続) [旧]:(無) [用]:(無)
こわいろ	こわいろ (基本形)-2(名詞) [旧]:こわいろ 0 [用]:(無)
を	を (基本形)-61(助詞) [旧]:(無) [用]:(無)
使っ	使う (基本形)-47-21-5(動詞-五段・ワ行促音便-連用夕接続) [旧]:(無) [用]:(無)
て	て (基本形)-64(助詞) [旧]:(無) [用]:(無)

[出力]

しわを伸ばしたやうなこわいろを使つて

図 3 解析結果トレース

## 3 インストール

### 3.1 本体, CGI モジュール

misima 本体及び Web CGI 環境の導入について述べる。%, #はそれぞれ Tcsh の一般ユーザ, スーパーユーザのコマンドプロンプトを示す。

#### 3.1.1 Makefile の修正

添付の Makefile (図 4・14 頁) の指定値を修正する。変数, 指定値内容を表 6 (15 頁) に示す。できるだけ変更しないことが望ましい。

```

PREFIX      = /usr/local
WEBP        = $(PREFIX)/www/apache22
WEBDIR      = $(WEBP)/data/misima
CGIDIR     = $(WEBP)/cgi-bin
ETCDIR     = $(PREFIX)/etc
MISIMADIR  = $(ETCDIR)/misima
MISIMATMP  = /tmp/misima
MISIMALOG  = /var/log/misima.log
HOST       = http://url.of.your.site
FORM       = /misima/misima.html
EMACS      = $(PREFIX)/bin/emacs
ELISPPATH  = $(PREFIX)/share/emacs/site-lisp
MAKEDA     = $(PREFIX)/libexec/chasen/makeda
CHADICDIR  = $(PREFIX)/lib/chasen/dic/ipadicutf8

```

図 4 Makefile

#### 3.1.2 茶筌 UTF-8 辞書の作成

CHADICDIR 指定のパスに茶筌 UTF-8 辞書を作成する。ipadic-2.7.0 を展開したディレクトリ (ここでは ~/tmp/ipadic-2.7.0 として説明する) にある辞書ソースを UTF-8 変換して作成する。

以下に手順を示す。EUC-UTF-8 変換に `nkf` を用いている。

```

% cd ~/tmp/ipadic-2.7.0
% su -m
# setenv CHADICDIR /usr/local/lib/chasen/dic/ipadicutf8

```

```
# foreach i (*.cha *.dic)
foreach? nkf -w $i > $CHADICDIR/$i
foreach? end
# cd $CHADICDIR
# `chasen-config --mkchadic`/makemat -i w
parsing grammar.cha
parsing cforms.cha
...
tri-gram: . 24576
matrix size: 2446x2229 -> 362x304
# `chasen-config --mkchadic`/makeda -i w chadic *.dic
parsing grammar.cha
parsing cforms.cha
...
Symbol.dic
Verb.dic
379012 entries
325968 keys
# exit
```

表 6 Makefile 変数

変数	指定値
PREFIX	PREFIX
WEBP	Web トップディレクトリ
WEBDIR	misima Web リソース格納ディレクトリ
CGIDIR	misima CGI モジュール格納ディレクトリ
ETCDIR	ローカル管理ファイル格納ディレクトリ
MISIMADIR	misima 管理ファイル格納ディレクトリ
MISIMATMP	misima 処理一時ディレクトリ
MISIMALOG	misima ログファイル絶対パス名
HOST	misima CGI ホスト名 (サイト URL)
FORM	DocRoot からの misima フォームパス名
EMACS	Emacs プログラム絶対パス
ELISPPATH	Emacs Elisp サイトディレクトリ
MAKEDA	茶筌 <code>makeda</code> コマンド絶対パス
CHADICDIR	茶筌 UTF-8 辞書格納ディレクトリ

### 3.1.3 管理ファイルの作成

パッケージ添付の `env/misimarc`, `env/misima.conf.sample`, `env/chasenutfrc` を環境, ユーザ運用に応じて修正する.

#### 1. `misimarc`

`misima` の辞書環境, マーキング文字列などを定義するリソースファイルである. ユーザ辞書を使う場合は `UDIC` パラメータに辞書の絶対パスを指定する. 本ファイルは `make` によって `$HOME/.misimarc` にインストールされる. 必要に応じてこれをユーザのホームディレクトリにコピーしておく.

#### 2. `misima.conf.sample`

Web CGI で参照する管理ファイルである. 本ファイルは `make` によって `/usr/local/etc/misima/misima.conf` にインストールされる. `Makefile` で `WEBP`, `HOST`, `FORM` を指定していれば, `__URL__`, `__FRM__`, `__WEB__` は自動で設定されインストールされるので修正は不要である.

```
# Site URL
$mysite    = "__URL__"; # HOST に合わせる
# Form html file
$myform    = "__FRM__"; # FORM に合わせる
# Additional Auth sites
$authsites = "__URL__"; # HOST に合わせる
# misima web data dir (system path)
$webdir    = "__WEB__"/data/misima"; # WEBP に合わせる
```

`$authsites` はリファラーのチェック時に参照する値である. `misima` CGI は, `$authsites` にマッチしないリファラーをもつ要求に対する処理は行わない仕様になっている. `$authsites` に `$mysite` と異なる URL を指定すると, `$mysite` と `$authsites` とが実行許可対象とされるフォーム・サイトとなる.

#### 3. `chasenutfrc`

茶筌が参照する辞書や動作を定義するファイルである. 次のパス指定を `CHADICDIR` (`ipadic UTF-8` 辞書インストール・ディレクトリ) に修正する. `ipadic` オリジナルの `EUC-JP` 辞書とは別に `UTF-8` 辞書用ディレクトリを作成することをお勧めする.

```
(GRAMMAR /usr/local/lib/chasen/dic/ipadicutf8)
```

本ファイルは `make` によって `/usr/local/etc/chasenutfrc` にインストールされる.



### 3.1.4 インストールの実行

スーパーユーザ権限で `make install` を実行すると、misima 本体・辞書、CGI モジュール環境がインストールされる。1.6 節 (7 頁) に記述した諸事項がすでに完了していなければならない。

```
# make install
```

ターゲット名 `misima-install`、`misima-web-install` によって、それぞれ misima 本体・辞書、CGI モジュール環境を個別にインストールできる。ローカルでコマンドライン利用するだけならば前者を実行するだけでよい。

```
# make misima-install
# make misima-web-install
```

### 3.1.5 環境変数の登録

`CHASENRC` 環境変数に `chasenutfrc` ファイルパスを登録する。以下のパス設定は本パッケージ・インストールの標準時のものである。

- `Tcsh` の場合:

```
% setenv CHASENRC /usr/local/etc/chasenutfrc
```

- `Bash` の場合:

```
$ export CHASENRC=/usr/local/etc/chasenutfrc
```

なお本登録を行うと、`CHASENRC` 標準の EUC-JP 辞書が参照されなくなるので注意のこと。茶筌標準の辞書を参照して `chasen` コマンドを利用する場合は、`chasen -r /usr/local/etc/chasenrc` などのように標準提供の `chasenrc` ファイルを明示指定して実行する。

## 3.2 Emacs Lisp (Elisp)

Emacs 用 Elisp のインストールは以下のとおり。

```
# make misima-elisp-install
```

### 3.3 SOAP Webservice

SOAP (Simple Object Access Protocol) は Web サービス・オブジェクトを送受信するためのメッセージ規約である。<sup>\*20</sup> SOAP Webservice は Web の HTTP プロトコル上での XML (eXtensible Markup Language) データの送受信に基づく標準であって、Web ブラウザ、HTML データ形式を前提としない。Web ブラウザをほぼクライアントの前提とする Web CGI とは異なり、アプリケーションを限定せず、ローカル PC 上のコマンドラインや Emacs からリモートにあるサーバ上の misima を呼び出して変換サービスを実現することができる。

本 misima パッケージは misima 本体を SOAP Webservice 通信経由でリモート PC から利用するためのサーバ `misimaSoapConnector`、Java クライアント `misimaSoapClient` 及び、Microsoft Word から変換を行うマクロテンプレートを同梱している。

本節では SOAP Java リソースのインストール方法について示す。SOAP サーバ環境構築が不要ならば読み飛ばしてよい。Microsoft Word クライアントについては 7.4 節 (42 頁) を参照。

#### 3.3.1 前提環境の準備

Java, Tomcat, Axis 及びその前提となるクラスライブラリの導入・環境設定を事前に実施しておく。本説明ではこれらがすでに完了しているものとする。Tomcat, Axis の導入についてはそれぞれ、<http://tomcat.apache.org/>、<http://ws.apache.org/axis/> を参照のこと。

#### 3.3.2 インストール Makefile の修正

`Makefile` 中の変数値 (図 5・19 頁) を、サーバ環境に合わせて修正する。

変数値の設定内容を表 7 (19 頁) に示す。

`SOAPURL` 値は SOAP リクエスト・エンドポイントを指定する。提供リソースでは筆者のサイトの URI (<http://yasuda.homeip.net/axis/services/misimaSoapConnector>) になっている。独自に SOAP サーバを構築する場合は運用に合わせて必ず修正しなければならない。本指定値によってクライアント実行スクリプト `soap/tools/misima.sh` 20 行目の URI 変数値を書き換える仕掛けになっている。利用者が `URI` 値の URI を直接書き換えてもよい。

```
# - URI 変数に Webservice 接続先を設定すること。
URI=http://yasuda.homeip.net/axis/services/misimaSoapConnector
```

<sup>\*20</sup> SOAP 仕様については W3C サイト <http://www.w3.org/TR/SOAP/> を参照のこと。SOAP に関する詳細な解説は文献 [8, 9]などを参照のこと。

```
JDKBINDIR    = /usr/local/jdk1.5.0/bin
AXISDIR      = $(PREFIX)/tomcat5.5/webapps/axis/WEB-INF
DEPLOYDIR    = $(AXISDIR)/classes
LOGPRPTDIR   = $(DEPLOYDIR)
SOAPURL      = $(HOST)/axis/services/misimaSoapConnector
SVCPORT      = 80
SOAPLOG      = /var/log/misima_soap.log
MISIMAJARDIR = $(HOME)/lib
USR          = www
GRP          = www
```

図 5 Makefile SOAP 関連変数

表 7 SOAP 用変数

変数	指定値
JDKBINDIR	java, javac など JDK 実行バイナリのディレクトリ
AXISDIR	Axis WEB-INF のディレクトリ
DEPLOYDIR	WebService をデプロイ (配備) するディレクトリ
LOGPRPTDIR	Log4J ロギングプロパティファイルを設置するディレクトリ
SOAPURL	misima SOAP WebService の URI
SVCPORT	misima SOAP WebService のポート番号
SOAPLOG	SOAP ログファイルの絶対パス
MISIMAJARDIR	SOAP クライアント jar ファイルの格納ディレクトリ
USR	Web サービス実行オーナ名
GRP	Web サービス実行グループ名

SOAPLOG を運用都合で変更する場合は、log4misima.properties 内にあるログファイル名も同期させる必要がある。本 Makefile は自動的に log4misima.properties を書き換えてインストールするようになっている。

### 3.3.3 Java モジュールのコンパイル

make によってサーバモジュール、クライアントモジュールをコンパイルする。クライアントについてはすでにコンパイル済み soap/lib/misima.jar を添付しているので必ずしも実施する必要はない。

```
% make misima-soap-server
% make misima-soap-client
```

Java コンパイルには、表 8 (20 頁) に示す非標準のクラスライブラリが必要である。インターネットリソースから取得してクラスパスに登録しておく。

表 8 必要クラスライブラリ

ライブラリ	サーバ	クライアント	備考
<a href="#">axis.jar</a>	○	○	Axis 1.4 添付
<a href="#">jaxrpc.jar</a>	○	○	Axis 1.4 添付
<a href="#">log4j-1.2.8.jar</a>	○		Axis 1.4 添付
<a href="#">servlet-api.jar</a>	○		Tomcat 5.5 添付
<a href="#">java-getopt-1.0.13.jar</a>		○	GNU getopt Java port <sup>*1</sup>

<sup>\*1</sup> <http://www.urbanophile.com/arenn/hacking/download.html>

### 3.3.4 Java ドキュメントの生成

`javadoc` によって Java プログラム・ドキュメントを生成する。

```
% make misima-soap-docs
```

### 3.3.5 クラスのインストール/サーバ環境設定 (デプロイ)

SOAP 関連リソースを所定の位置に格納するとともに、`misimaSoapConnector` (SOAP サーバ) のデプロイ (配備) を実行する。デプロイは Axis の `AdminClient` によって行われる。これとともに、`misima SOAP` モジュールの Java プログラム・ドキュメントが `WEBDIR` に指定したディレクトリにインストールされる。スーパーユーザ権限が必要である。

```
# make misima-soap-install
```

misimaSoapConnector をアンデプロイ (削除) したい場合, 添付の WSDD (Web Service Deployment Descriptor) `undeploy-misimaconnector.wsdd` を用いて `AdminClient` を次のように実行すればよい.

```
# cd /usr/local/tomcat5.5/webapps/axis/WEB-INF/classes
# java org.apache.axis.client.AdminClient -p 80 \
  undeploy-misimaconnector.wsdd
```

再度デプロイしたい場合, `deploy-misimaconnector.wsdd` を用いる. サービス URI は運用に合わせて変更する. `AdminClient` の実行方法詳細については Axis 添付のマニュアルを参照のこと.

```
# cd /usr/local/tomcat5.5/webapps/axis/WEB-INF/classes
# java org.apache.axis.client.AdminClient -p 80 \
  -lhttp://yasuda.homeip.net/axis/service/misimaSoapConnector \
  deploy-misimaconnector.wsdd
```

### 3.3.6 Tomcat 環境変数の設定

misimaSoapConnector は `MISIMA_LOG_PROPERTIES_PATH` 環境変数から Log4J ログインプロパティファイル (`log4misima.properties`) の絶対パス名を取得する. このため Tomcat サブレットエンジンの環境変数にこれをセットしておく必要がある.

`Makefile misima-soap-install` レシピでインストールすると, `log4misima.properties` は `$DEPLOYDIR` 指定値の示すディレクトリ下にインストールされる. 環境変数の値にはこの絶対パス名を指定する.

Tomcat の起動用スクリプト (`$CATALINA_HOME/bin/catalina.sh` 等) に以下のように設定する. コマンドラインから手動で起動スクリプトを発行する場合, その直前で同一の仮想端末から環境変数をセットしてもよい.

```
MISIMA_LOG_PROPERTIES_PATH=log4misima.properties 絶対パス名
export MISIMA_LOG_PROPERTIES_PATH
```

上記が完了したら, Tomcat を再起動する. これで misima SOAP サーバが利用可能となる.

Axis の管理画面でデプロイが正常に完了しているかを確認する. まず `http://host/axis/` にアクセス

し、管理トップ画面を表示させる (図 6・22 頁). \*21 ここから「リスト」をクリックすると図 7 (22 頁) が出力される。この一覧に“misimaSoapConnector (wsdl)”が表示されていればデプロイが完了している。“(wsdl)”をクリックすると WSDL が自動生成され、図 8 (22 頁) のとおり表示されるはずである。

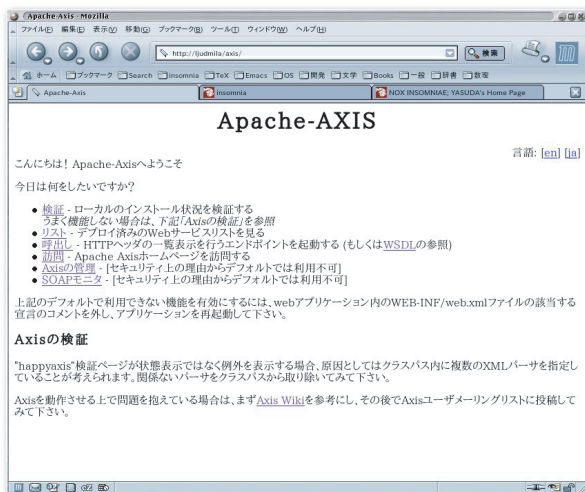


図 6 Axis 管理画面トップ

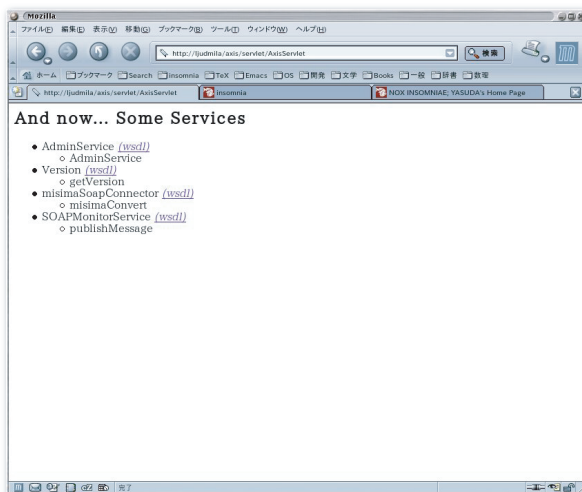


図 7 Axis サービス一覧

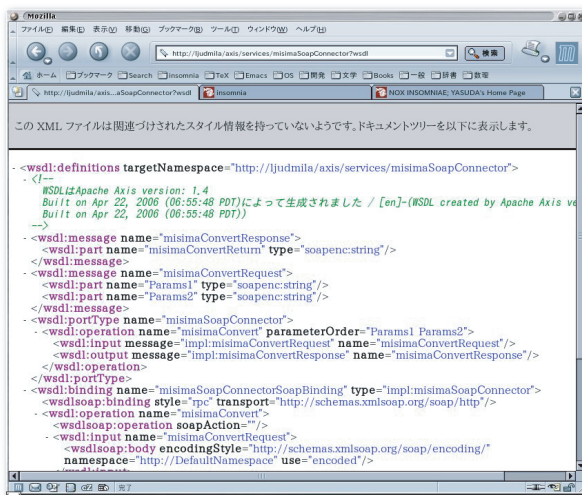


図 8 misimaSoapConnector 自動生成 WSDL

### 3.3.7 Emacs SOAP 用 ELisp のインストール

Emacs から misima SOAP Webservice を利用するための ELisp をインストールする。

\*21 本稿では Axis Web サービスのポート番号を 80 として説明している。Tomcat の標準設定ではポート番号を `http://host:8080/axis` などと指定しなければならない。ポート番号を再設定するには Tomcat の管理ファイルである `server.xml` の `Connector` 定義、もしくは Apache22 の `proxy_ajp` プロキシ定義で変更しておく。詳細は Tomcat 利用バージョン、Apache22 `proxy_ajp` モジュールのマニュアルを参照のこと。

```
# make misima-soap-elisp-install
```

## 4 misima コマンド

misima の実行は、仮想端末のシェルから `misima` コマンドを発行するのが基本である。標準入力から UTF-8 テキストを読み込み、標準出力に変換後の UTF-8 テキストを、標準エラー出力にエラー情報を書き出す。ファイルの入出力についてはリダイレクトの使用を想定している。

図 9 (24 頁) にコマンドライン misima 実行例を示す。

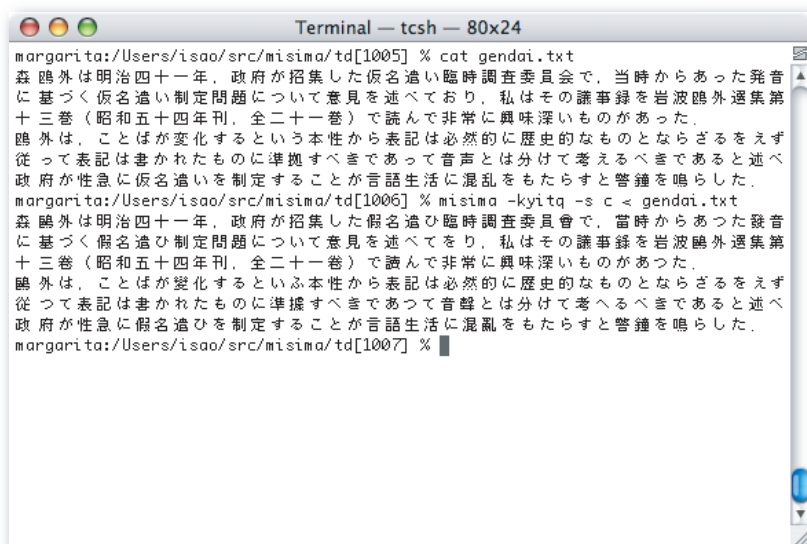


図 9 コマンドライン操作

### 4.1 コマンドライン書式

`misima` コマンドラインの指定書式を図 10 (24 頁)、引数の意味を表 9 (28 頁) に示す。

```
misima [ -k|-y|-t|-i|-n|-q|-d|-s {cjhuam}|-x <k|u|f|x|r|t|T|h|i|a>|
-c {KCT}|-m {ht}|-r "rcfile" ]
```

- [ ] 内は省略可能である。
- | で区切られたパラメータは同時指定が可能である。
- { } 内は前置オプションが指定されたとき、いずれかを必ず指定する。
- < > 内は前置オプションが指定されたとき、いずれかを必ず指定する（複数可）。

図 10 misima コマンドライン書式



## 4.2 .misimarc リソースファイル

`.misimarc` リソースファイルは `misima` 辞書のパス情報、マーキングのための文字列定義を管理している。提供ファイル `env/misimarc` を変更せずにインストールした場合、`.misimarc` はシステムの標準値が設定されている。

`misima` はデフォルトでは `$HOME/.misimarc` を参照する。コマンドライン・オプション `-r rcfile` で指定したリソースファイルを参照するよう `misima` に指示することができる。

標準値を変更したい場合は、`$HOME/.misimarc` もしくは別途新規にリソースファイルを作成してその定義内容を変更する。

パラメータと指定値を空白文字で区切って対で記述する。`HTMLTGD`、`HTMLTGS`、`TEXCMD`、`TEXCMDS` の指定値については、マーキングタグ/コントロールシーケンスの開始及び終了の文字列を、空白で区切って指定しなければならない。行頭に“`#`”をタイプすると当該行をコメントとして無視する。

標準インストールの `$HOME/.misimarc` を図 11 (25 頁) に、各種パラメータの意味を表 10 (29 頁) に示す。

```
HTMLTGD <font color="Red"> </font>
HTMLTGS <font color="Blue"> </font>
MOJIKYO \MO
TEXCMD \red{ }
TEXCMD \blue{ }
SDIC    /usr/local/etc/misima/sdic
JISTBL  /usr/local/etc/misima/jis-tbl
EUROTBL /usr/local/etc/misima/euro-tbl
T2ATBL  /usr/local/etc/misima/t2a-tbl
OCSTBL  /usr/local/etc/misima/ocs-tbl
EXT_DIR /usr/local/etc/misima/contrib
# ユーザ辞書を使う場合はパスを指定し、先頭 # を削除する。
#UDIC   /Users/isao/src/misima/dic/udic
```

図 11 `.misimarc`

## 4.3 mlatex ユーティリティ

`mlatex` コマンドは UTF-8 で準備した  $\text{T}_\text{E}\text{X}$  原稿を対象に `misima` 変換、 $\text{p}\text{E}\text{T}_\text{E}\text{X}$  コンパイル、`dvipdfmx` PDF 生成を実行するユーティリティである。本ユーティリティは、現代仮名遣い・常用漢字で入力した  $\text{T}_\text{E}\text{X}$  文書の `misima` 変換手続きを簡略化するために用意されている。

`Makefile` ではインストールされない。`module/mlatex` をカスタマイズし、実行パスの通った場所に格

納して利用する。TEXRC は予め \$HOME/.misimarc をコピーするなどして作成しておく。  
 pTeX 一式, dvipdfmx, iconv 外部プログラムが必要である。  
 mlatex の変数, 指定内容をそれぞれ図 12 (26 頁), 表 11 (29 頁) に示す。

```
JPCODE=EUC-JP
TEXRC=/usr/local/etc/misima/texrc
OPT="-gtkyl -s a -x kifxt -r $TEXRC"
PLATEX=$PREFIX/teTeX/bin/platex
DVIPDFMX=$PREFIX/teTeX/bin/dvipdfmx
```

図 12 mlatex 変数

mlatex のコマンドライン書式は図 13 (26 頁) のとおりである。dvipdfmx option は省略可能である。tex-file は拡張子 .tex を省略できる。mlatex はログメッセージをチェックして、TeX 原稿にある \ref 等のリファレンスが解決するまで platex コンパイルを自動実行するようになっている。

```
mlatex [dvipdfmx option] tex-file
- dvipdfmx option: dvipdfmx に渡すオプション (省略可)
- tex-file: 入力ファイル (拡張子省略可)
```

図 13 mlatex コマンドライン書式

## 5 Emacs からの変換

### 5.1 .emacs の設定

misima を GNU Emacs から利用するために、Emacs の個人用初期設定ファイルである .emacs に図 14 (27 頁) に示す記述を追加する。

### 5.2 Emacs コマンド

Emacs からの misima 実行は“M-x misima-コマンド RET”とする。-region 名のコマンドはリージョン (対象テキスト範囲) を設定して実行するものである。

表 12 (30 頁) にコマンド一覧を示す。

misima Elisp を用いると、Emacs で文書作成をしつつ必要なテキスト領域に対して misima 変換を施すことができる。図 15 (27 頁), 及び図 16 (27 頁) に misima Emacs Lisp の変換前後のスクリーンショット (Mac OS X Tiger, X11-Emacs 22.0.50) を示す。

```
;; misima.elc の指定
(require 'misima)
;; misima プログラムの絶対パス指定
(setq misima-path "/usr/local/bin/misima")
;; misimarc の絶対パス指定 ($HOME/.misimarc)
(setq misima-rc "/home/yukio/.misimarc")
```

図 14 .emacs

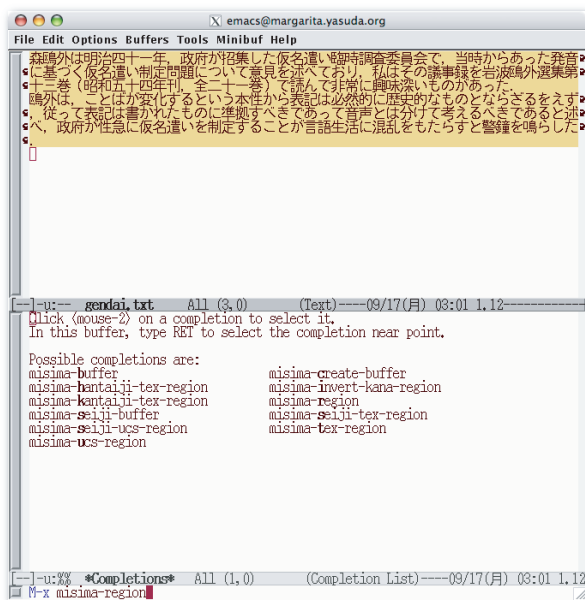


図 15 Emacs 操作 (変換前)

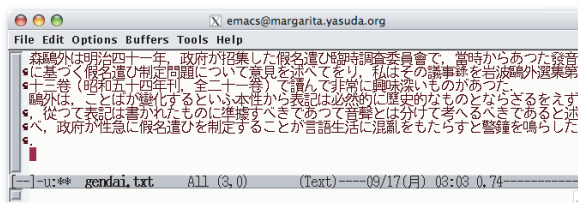


図 16 Emacs 操作 (変換後)

表 9 misima コマンドライン引数

引数	意味
-k	旧仮名変換を行う
-y	用字・用語変換を行う
-t	単純変換を行う
-i	繰返し符号変換を行う
-n	仮名反転変換を行う
-s	引数により旧字変換を行う
c	UTF-8 コード
j	JIS 第一・第二水準範囲内
h	HTML 数値参照
u	TeX OTF package UTF-8 数値参照
a	TeX OTF package CID 参照
m	TeX 今昔文字鏡大漢和文字番号参照
-x	引数により TeX 変換を行う
k	漢文訓点変換
u	多言語標準変換
f	多言語強制変換
x	多言語拡張変換
r	Cyrillic T2A 変換
t	タイ語変換 (TIS-620 十六進数変換有)
T	タイ語単語 (UTF-8 出力)
h	UTF-8 十六進数変換
i	くノ字点変換
a	kui に同じ
-c	JIS 中国語・韓国語変換を行う
K	\UTFK (韓国語)
C	\UTFC (中文簡体字)
T	\UTFT (中文繁体字)
-m	マーキングを行う
h	HTML タグ出力
t	TeX 命令出力
-q	致命的エラー以外のメッセージを抑止する
-d	形態素解析結果トレースを出力する
-r	リソースファイルを指定する
rcfile	リソースファイル名

表 10 リソースファイル・パラメータ

パラメータ	意味
HTMLTGD	HTML 辞書マーキングテキスト: 開始 終了
HTMLTGS	HTML 本則マーキングテキスト: 開始 終了
TEXCMDD	TeX 辞書マーキングテキスト: 開始 終了
TEXCMDS	TeX 本則マーキングテキスト: 開始 終了
MOJIKYO	今昔文字鏡コントロールシーケンス
SDIC	システム辞書絶対パス
JISTBL	JIS テーブル絶対パス
EUROTBL	多言語変換テーブル絶対パス
T2ATBL	Cyrillic T2A 変換テーブル絶対パス
OCSTBL	教会スラヴ語変換テーブル絶対パス
EXT_DIR	拡張変換表格納ディレクトリ絶対パス
UDIC	ユーザ辞書絶対パス (使用する場合のみ指定)

表 11 mlatex 変数指定値

変数	指定値
JPCODE	pLaTeX 日本語文字コード
TEXRC	mlatex 用リソースファイル, <code>.misimarc</code> コピーでも可
OPT	misima コマンドライン・オプション \$TEXRC 不要の場合 <code>-r \$TEXRC</code> は削除する
PLATEX	platex 絶対パス
DVIPDFMX	dvipdfmx 絶対パス

表 12 Emacs コマンド

コマンド	機能
<code>misima-region</code>	<code>= -ykitq<sup>*1</sup></code>
<code>misima-ucs-region</code>	<code>= -ykitq -s c</code>
<code>misima-tex-region</code>	<code>= -ykitq -s a -x a</code>
<code>misima-seiji-ucs-region</code>	<code>= -q -s c</code>
<code>misima-seiji-tex-region</code>	<code>= -q -s a -x u</code>
<code>misima-invert-kana-region</code>	<code>= -nq</code>
<code>misima-kantaiji-tex-region</code>	<code>= -q -x u -c C</code>
<code>misima-hantaiji-tex-region</code>	<code>= -q -x u -c T</code>
<code>misima-buffer</code>	<code>= -ykitq -s c<sup>*2</sup></code>
<code>misima-seiji-buffer</code>	<code>= -qs c<sup>*2</sup></code>
<code>misima-create-buffer</code>	<code>= -ykitq -s c<sup>*3</sup></code>

\*1 `-s` オプション問い合わせ

\*2 当該バッファ

\*3 新規バッファ

## 6 misima Web 環境

misima を Web 環境で運用するための設定内容、利用方法について述べる。ユーザインタフェースであるフォーム HTML とサーバ側の CGI モジュールから構成される。前者は運用管理者が misima の CGI パラメータ仕様に準拠しつつカスタマイズすることを想定している。

### 6.1 フォームのカスタマイズ

Web CGI を運用するにあたり、`html/` 下にある `misima.html`、`misima-header`、`misima-footer` をカスタマイズする。

`misima.html` は CGI サブミット・フォーム用 HTML のプロトタイプである。`misima-header`、`misima-footer` は misima CGI 変換出力結果の前後に付加される HTML コードである。

これらの提供リソースは筆者のサイト <http://yasuda.homeip.net/> のコードそのものであり、画像や JavaScript 等のパスは当サイト環境に依存している。利用者の環境に応じてカスタマイズ・修正が必要である。HTML ソース修正は容易であると考え、変更箇所等の具体的な説明は割愛する。

CGI パラメータ仕様を表 13 (32 頁) に示す。ここで、指定値 10 は 1 または 0 のいずれかが指定可能で、機能のオン (1) | オフ (0) を示す。text は UTF-8 テキスト文字列を示す。u|r|t|T|... は複数選択可能指定、CTK0 はいずれかの文字ひとつ選択で、0 の場合はオフを示す。それぞれの変換機能の意味は表 9 (28 頁) を参照のこと。

指定の実際は `misima.html` のソースコードを参考にしていただきたい。

### 6.2 misima.conf

`$_MISIMADIR/misima.conf` には Web CGI モジュール `misima.cgi` の動作を制御するパラメータが定義されている。

本パラメータは Perl 文法に準拠して指定しなければならない。HTML コードを記述する場合には文字のエスケープに注意する。`misima.cgi` によって実行 (評価) される Perl コードそのものであり、予め記述されているパラメータへの設定以外の Perl コードを決して記述してはならない。`misima.conf` に不正な記述が存在した場合の misima Web CGI モジュールの動作は保証されない。

`misima.conf` 各種パラメータの意味を表 14 (33 頁) に示す。

### 6.3 misima Web 変換フォームの使い方

misima フォームの使い方を <http://yasuda.homeip.net/misima/misima.html> で説明する。

#### 1. フォームへのアクセス

ブラウザで <http://yasuda.homeip.net/misima/misima.html> を開く。

#### 2. テキストの入力・オプションの指定

変換対象としたいテキストを「テキスト入力」のエリアに入力する。「旧仮名遣い変換」、「旧字変換」などの変換オプションを選択して「変換」ボタンをクリックする。図 18 (34 頁) では旧仮名遣い変換を「する」に、旧字変換出力形式を「UTF-8」に指定する例である。

表 13 CGI パラメータ

パラメータ	指定値	内容 (相当する <code>misima</code> コマンド引数)
<code>itext</code>	<code>text</code>	変換対象テキスト
<code>k_opt</code>	<code>10</code>	旧仮名変換 ( <code>-k</code> )
<code>t_opt</code>	<code>10</code>	単純変換 ( <code>-t</code> )
<code>s_opt</code>	<code>cjhua0</code>	旧字変換 ( <code>-s {cjhua}</code> )
<code>y_opt</code>	<code>10</code>	用字・用語変換 ( <code>-y</code> )
<code>i_opt</code>	<code>10</code>	繰返し符号変換 ( <code>-i</code> )
<code>n_opt</code>	<code>10</code>	仮名反転変換 ( <code>-n</code> )
<code>x_opt</code>	<code>u r t T x f k i</code>	$\TeX$ 変換 ( <code>-x &lt;u r t T x f k i&gt;</code> )
<code>c_opt</code>	<code>CTK0</code>	$\TeX$ 中韓変換 ( <code>-C {CTK}</code> )
<code>d_opt</code>	<code>10</code>	解析結果出力 ( <code>-d</code> )
<code>m_opt</code>	<code>ht0</code>	マーキング用指定 ( <code>-m {ht}</code> )
<code>m_opt_htds</code>	<code>text</code>	HTML 辞書マーキング開始用テキスト
<code>m_opt_htde</code>	<code>text</code>	HTML 辞書マーキング終了用テキスト
<code>m_opt_htss</code>	<code>text</code>	HTML 本則マーキング開始用テキスト
<code>m_opt_htse</code>	<code>text</code>	HTML 本則マーキング終了用テキスト
<code>m_opt_txds</code>	<code>text</code>	$\TeX$ 辞書マーキング開始用テキスト
<code>m_opt_txde</code>	<code>text</code>	$\TeX$ 辞書マーキング終了用テキスト
<code>m_opt_txss</code>	<code>text</code>	$\TeX$ 本則マーキング開始用テキスト
<code>m_opt_txse</code>	<code>text</code>	$\TeX$ 本則マーキング終了用テキスト
<code>s_mojikyo</code>	<code>text</code>	今昔文字鏡番号指定 $\TeX$ コマンド
<code>utext</code>	<code>text</code>	ユーザ辞書テキスト

### 3. 旧字変換オプションの注意事項

旧字変換オプションで UTF-8 出力を選択していると、貼付け先のアプリケーションできちんと文字が表示されない場合がある。これは当該アプリケーションが Unicode を扱えないためである。ブログ管理システムが SJIS エンコードで設定されていたりすると、投稿した記事において Unicode 固有の文字は「?」に化けてしまう。この場合、旧字変換オプションに「**JIS 第一・第二水準**」を選択すれば、JIS 第一・第二水準の範囲内で旧字体を出力するので、問題を回避できる。ただし、対応する旧字体が Unicode にしかない新字は変換されないまま出力されることに留意すべきである。

### 4. $\TeX$ 変換

変換結果を  $\TeX$  で活用するならば、「 **$\TeX$  変換**」オプションから必要な機能を選択する。Adobe-Japan1.5 の広範囲の文字空間から旧字体を拾うことができる。

### 5. ユーザ辞書の指定

ユーザ辞書を記述するテキストエリアにユーザ独自の定義を記述すると、変換結果に反映させることができる。定義の様子は 9 節 (49 頁) を参照のこと。「**保存**」ボタンをクリックするとユーザ定義が



表 14 misima.conf パラメータ

パラメータ	意味
\$mysite	サイト URL
\$myform	フォームのパス
\$authsites	許可 URL (空白で区切って複数可)
\$work_dir	一時ワークディレクトリ
\$misima	misima 絶対パス
\$logfile	ログファイル
\$maxtxt	入力テキスト文字数最大値
\$sdic	システム辞書絶対パス
\$jistbl	JIS テーブル絶対パス
\$eurotbl	多言語変換テーブル絶対パス
\$misima_v	公開バージョン
\$misima_d	公開日付
\$cssfile	CSS ファイル URL
\$webdir	misima Web リソースディレクトリ
\$html_header	ヘッダコード絶対パス
\$html_footer	フッタコード絶対パス
\$misima_opt_pre	misima オプション出力前置コード
\$misima_opt_aft	misima オプション出力後置コード
\$pre_text	misima 変換結果前置コード
\$aft_text	misima 変換結果後置／入力テキスト前置コード
\$end_text	入力テキスト後置コード
\$unauthmes	許可 URL 以外からの要求時のエラーメッセージ

クッキーとしてブラウザに記憶させることができる。「読込」ボタンをクリックすることで、保存した内容呼び出すことができる。フォームではユーザ辞書エリアには既定値が設定されている(図 17・34 頁)。これは一部康熙字典別掲漢字・人名漢字を変換しない定義を行ったものである。UNIX, Windows のバージョンではこれらの漢字が表示されない場合があるための措置である。Mac OS X Safari や Windows XP Internet Explorer Ver.7 では表示できるので、この初期値をクリアすることをお勧めする。

#### 6. 変換の実行と結果の出力

「変換」ボタンを押し下げて、しばらくすると図 19 (34 頁) のような結果が表示される。この変換結果をコピーして文書に貼付けて活用する。

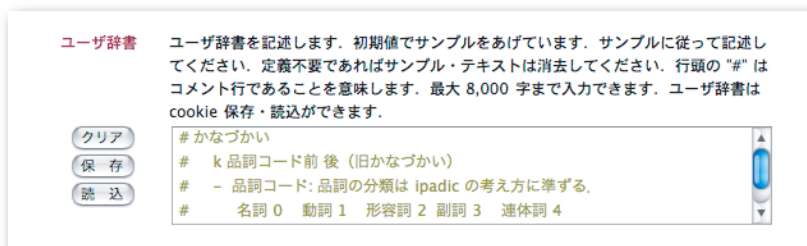


図 17 変換結果出力

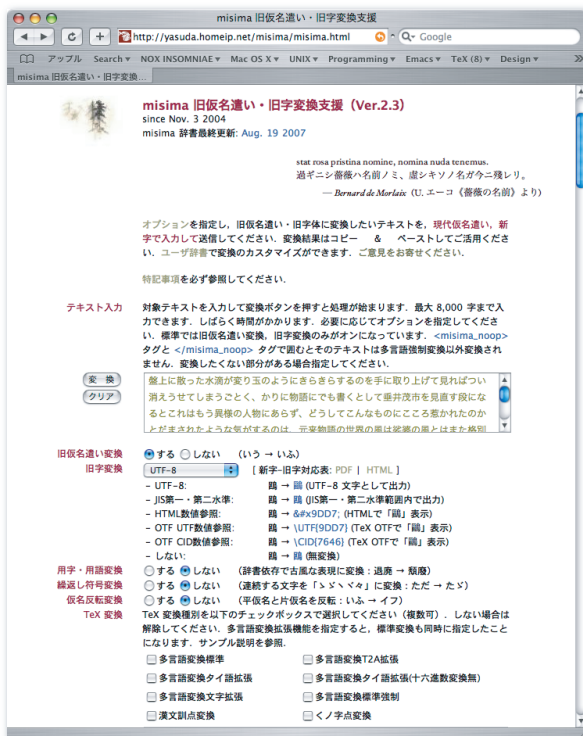


図 18 フォーム入力 (変換前)



図 19 変換結果出力

## 7 SOAP Webservice

本節では misima SOAP Java クライアント `misimaSoapClient.class` 及び Microsoft Word クライアントをセットアップする方法と、その使い方について述べる。

misima SOAP サーバ環境は Web サブレット環境へのデプロイまで `Makefile misima-soap-install` レシピで整うようになっている。サーバ運用は Tomcat, Axis のマニュアルを参照して運用者の都合に合わせて設定する。

misima SOAP クライアントは、`misimaSoapConnector` の SOAP インタフェースに準拠すれば、ユーザ独自に作成することができる。パッケージには SOAP サービスインタフェースを表した WSDL (Web Services Description Language) `soap/conf/misimaSoapConnector.wsdl` を添付している。これは Axis `Java2WSDL` で自動生成したものである。SOAP サーバがデプロイ済みならば、ブラウザから `http://host/axis/services/misimaSoapConnector?wsdl` にアクセスしても WSDL が得られる。これを Axis 付属のユーティリティ `WSDL2Java` や Microsoft Office XP Web Services Toolkit で処理すれば、misima SOAP クライアント作成のための Java/VBA スタブクラスコードが生成される。利用者独自クライアント作成の労力が軽減されると思われる。

### 7.1 Java クライアント環境

#### 7.1.1 OS/Emacs 環境

Windows 2000/XP, Mac OS X, UNIX 系 OS (FreeBSD) の各環境で動作する。Linux, Windows 95/98 や Max OS Classic は確認できていない。筆者は Windows 2000 SP4, Windows XP SP2, Mac OS X 10.4.10 Tiger, FreeBSD 6.2-RELEASE で開発/動作確認を行っている。

本パッケージではコマンドラインからアクセスするツール (Java クラスライブラリ, ラッパーシェルスクリプト, ラッパーバッチファイル) と、これを利用して Emacs から misima 機能を使う Emacs Lisp (ローカル用及び SOAP リモートサーバアクセス用) を添付している。

Emacs での利用は当然ながら Emacs (Windows の場合は Meadow) が適切に動作するように予めセットアップしておく必要がある。さらに旧字変換機能, TeX 多言語文字変換機能を利用するのなら, Emacs で Unicode フォントが利用できるようにしておく。Emacs バージョンは X11-Emacs 22.0.50 (Mac OS X), Meadow 20050617-1 版 (22.0.50) で確認している。Mac の Carbon-Emacs は未確認である。misima TeX 多言語文字変換の元テキスト編集で Unicode フォントを利用するとすれば, X11-Emacs を推奨する。

#### 7.1.2 Java 環境

JVM (Java Virtual Machine) が動作する Java 実行環境が必要である。Windows, Linux, FreeBSD については, JavaWorld などの雑誌の付録 CD-ROM を手に入れるなり, Sun のサイトからダウンロードするなりして, J2SE (Java 2 Platform, Standard Edition) 5.0 以上の JRE (Java Runtime Environment) もしくは SDK (Software Development Kit) をインストールしておく。Mac OS X の場合は Apple 提供メディアから Mac OS X Developer Tools をインストールする。いずれの場合も, クラスパスを適切に設定して `java` コマンド (クラスランチャ) が正常に動作するようになっていなければならない。Java 実行環境の設置につ

いては、JRE、SDK のドキュメント<sup>\*22</sup>やインターネット・リソースを参照のこと。

SOAP アクセス基幹部分は Java クラスライブラリ (`misima.jar`) である。これは標準の Java 実行環境のほかに表 15 (36 頁) に示すクラスライブラリを必要とする。別途インターネットリソースから入手し、クラスパスに登録しておく。

表 15 クライアント実行に必要なクラスライブラリ

ライブラリ	備考
<code>axis.jar</code>	Axis 1.4 添付
<code>jaxrpc.jar</code>	Axis 1.4 添付
<code>saaj.jar</code>	Axis 1.4 添付
<code>log4j-1.2.8.jar</code>	Axis 1.4 添付
<code>wsdl4j-1.5.1.jar</code>	Axis 1.4 添付
<code>commons-logging-1.0.4.jar</code>	Axis 1.4 添付
<code>commons-discovery-0.2.jar</code>	Axis 1.4 添付
<code>java-getopt-1.0.13.jar</code>	GNU getopt Java port <sup>*1</sup>

<sup>\*1</sup> <http://www.urbanophile.com/arenn/hacking/download.html>

なお、本文書の説明においては、Mac OS X を UNIX 系に含めて記述している。

## 7.2 SOAP Java クライアントのインストール

### 7.2.1 Windows

本パッケージにはすでに Java コンパイル済みの `misima.jar` が添付されている。これは J2SE 5.0 で作成したものであり、実行にはバージョン 5.0 以上の JRE が必要である。

Windows へのインストールでは、ファイルのコピー操作はエクスプローラから実行すればよいが、一部「コマンドプロンプト」画面からコマンドを入力する必要がある。「コマンドプロンプト」は Windows 2000 標準では、「スタート」-「プログラム」-「アクセサリ」-「コマンドプロンプト」で起動する。Windows のバージョンによっては「MS-DOS プロンプト」という名称の場合がある。

#### 1. インストール・フォルダの作成

`misima` SOAP クライアントのインストール・フォルダを作成する。本説明では `C:%misima` と仮定する。ユーザの事情に合わせて変更可能であるが、その場合は説明中のフォルダ名を読み替えて指定しなければならない。

コマンドプロンプトを起動して、次を入力する。「`C:%WINDOWS>`」というコマンド・プロンプトはお使いの Windows 環境によって異なる場合がある。

<sup>\*22</sup> <http://java.sun.com/j2se/1.5.0/ja/>

```
C:¥WINDOWS>mkdir c:¥misima
```

2. `misima.jar` のコピー

`soap¥lib¥misima.jar` を, `C:¥misima` 直下にコピーする.

3. バッチファイル `misima.bat` の修正

`soap¥tools¥misima.bat` の `MISIMAJARDIR` 変数を, パッケージを展開したインストール・フォルダ名に修正する. `C:¥misima` 直下に解凍した場合は不要である.

```
set MISIMAJARDIR=c:¥misima
```

4. URI の修正

`soap¥tools¥misima.bat` の `URI` 変数を, `misima SOAP サーバ (misimaSoapConnector)` の `URI` に修正する. パッケージ添付の `misima.bat` では, 筆者のサイトの `URI` になっている.

```
set URI=http://yasuda.homeip.net/axis/services/misimaSoapConnector
```

5. `misima.bat` のコピー

修正した `soap¥tools¥misima.bat` をインストール・フォルダにコピーする.

6. 実行パスの設定

インストール・フォルダを Windows 「システムのプロパティ」 - 「詳細設定」 タブの環境変数設定画面で Windows 実行パスに登録する. これは必須ではないが, どのカレント・フォルダからでも `misima.bat` スクリプトを実行できるようにするためである.

7. クラスライブラリのダウンロードと格納

表 15 (36 頁) のクラスライブラリを入手して, ローカルハードドライブ (ここでは `C:¥lib` とする) に格納する. すなわち, ブラウザで Axis アーカイブサイト [http://archive.apache.org/dist/ws/axis/1\\_4/](http://archive.apache.org/dist/ws/axis/1_4/) にアクセスして, `axis-bin-1_4.zip` をダウンロードし, 展開する. `lib` フォルダ内にある `jar` ファイルを `C:¥lib` にコピーする.

8. クラスパスの登録

Windows 「システムのプロパティ」 - 「詳細設定」 タブの環境変数設定画面で `CLASSPATH` 変数を編集する. ユーザ/システムどちらの環境変数設定でもよい. システムの場合はシステム全体に適用される. `CLASSPATH` 変数が存在しなければ「新規 (W)」で, すでに登録されていれば「編集 (I)」でクラスパスの設定編集を行う. “.;`C:¥misima¥misima.jar`; `C:¥lib¥axis.jar`;...”のように“;” (セミコロン) で区切って必要 `jar` ファイルの絶対パスを記述する (図 20・38 頁). 定義が完了したら, コマンドプロンプトを起動して, `set CLASSPATH` を発行して, 登録したファイル名が図 21 (38 頁) のようにきちんと表示されるか確認する.

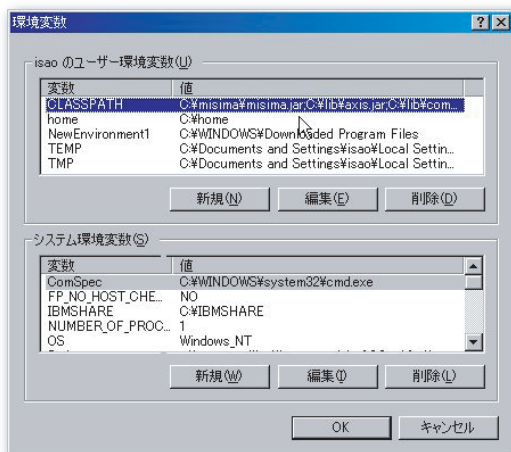


図 20 CLASSPATH 環境変数設定

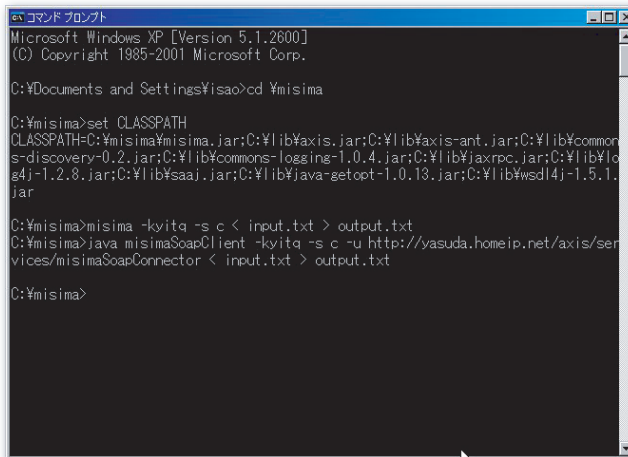


図 21 CLASSPATH 環境変数確認 / misima SOAP の実行

## 9. .emacs の設定

これ以降の手順はエディタ Meadow から misima SOAP サービスを実行する場合のみ必要である。Meadow 初期設定ファイル `.emacs` に以下の 2 行を追加する。 `misimasoap-path` は上記バッチファイルの絶対パス名を指定する。フォルダのセパレータは `"/` でよい。

```
(require 'misimasoap)
(setq misimasoap-path "c:/misima/misima.bat")
```

## 10. Emacs のインストール

添付 `misimasoap.el` を Emacs Lisp のロードパスの通ったフォルダにコピーする。Meadow をインストールしたフォルダにある `site-lisp` フォルダに格納すればよい。

### 7.2.2 UNIX

UNIX の場合、`Makefile misima-soap-install` 及び `misima-soap-elisp-install` レシピを実行すると、SOAP クライアント jar ファイル `misima.jar`、SOAP クライアント実行スクリプト `misima.sh` 及び Emacs Lisp `misimasoap.el` が所定のパスにインストールされる。必要クラスライブラリへの参照がクラスパスに設定されていれば SOAP クライアントはすぐにも動作する。

ここでは、手動でセットアップする手順を示す。

#### 1. jar ファイルの格納

misima SOAP クライアント jar ファイルを格納するディレクトリを作成し、`misima.jar` をコピーする。本説明ではインストール・ディレクトリ名を `/home/user/lib` と仮定して説明する。ユーザの事情に合わせて変更可能であるが、その場合は説明中のディレクトリ名を読み替えて指定しなければならない。

```
% cd ~
% pwd
/home/user
% mkdir -p lib
% cp -p tmp/misima-2.3i/soap/lib/misima.jar lib
```

## 2. シェル `misima.sh` の修正

`soap/tools/misima.sh` の `MISIMAJARDIR` 変数をインストール・ディレクトリ名に修正する。  
`URI` 変数に, misima SOAP サーバのエンドポイントを設定する。初期状態では筆者のサイト:<http://yasuda.homeip.net/axis/services/misimaSoapConnector> になっているので, 独自に構築した場合はその URI に再設定する。

```
MISIMAJARDIR=/home/user/lib
URI=http://yasuda.homeip.net/axis/services/misimaSoapConnector
```

## 3. `misima.sh` の格納

パスの通ったディレクトリに `misima.sh` を格納する。ここでは `/home/user/bin` とする。

## 4. `CLASSPATH` 環境変数の設定

`misima.jar` 及び表 15 (36 頁) に示す jar ファイルを `CLASSPATH` 環境変数に登録する。複数のパスは“:” (コロン) で区切って記述する。

## 5. `.emacs` の設定

以下の手順は Emacs から misima SOAP サービスを実行する場合のみに適用する。

Emacs 初期設定ファイル `.emacs` に以下の 2 行を追加する。`misimasoap-path` は `misima.sh` の絶対パス名を指定する。

```
(require 'misimasoap)
(setq misimasoap-path "/home/user/bin/misima.sh")
```

## 6. Emacs Lisp のインストール

添付 `misimasoap.el` をバイトコンパイルし, Emacs Lisp のロードパスの通ったディレクトリにコピーする。バイトコンパイルをせず `misimasoap.el` をそのままコピーしてもよい。

```
% cd misima-2.3i/soap/tools
% emacs -batch -f batch-byte-compile misimasoap.el
```

```
% sudo cp misimasoap.elc /usr/local/share/emacs/site-lisp
```

## 7.3 SOAP Java クライアントの実行

misima SOAP クライアントの実行形態は二つある。コマンドライン実行と Emacs 実行である。変換対象テキストの文字数は1回あたり 8,000 文字まで可能である。

### 7.3.1 SOAP コマンドライン

コマンドライン実行はラッパーツール (UNIX: `misima.sh`, Windows: `misima.bat`) を利用する方法と、`java` クラスランチャで `misimaSoapClient.class` を直接呼び出す方法である。前者は内部的に後者を実行しているだけである。

■ Windows コマンドプロンプトを起動して `misima.bat` を実行する。コマンド指定は `misima.bat` でも、単に `misima` でもよい。

```
C:¥MISIMA>misima -kyitq -s c < inputfile > outputfile
```

- “`-kyitq -s c`” は `misima` 実行オプションである。“`< inputfile`”を省略するとキーボードから入力する。`misima` は UTF-8 入力を前提としているためファイル入力として使うのが普通ではないかと思う。
- `inputfile` は変換元のテキストファイルである。
- `outputfile` は変換結果テキストファイルである。“`> outputfile`”を省略すると画面に出力する。

実行の様子は図 21 (38 頁) のはじめの実行例のとおりである。

■ UNIX 仮想端末から `misima.sh` を実行する。シェルスクリプトである以外は Windows と同じである。

```
% misima.sh -kyitq -s c < inputfile > outputfile
```

■ SOAP コマンドライン・オプション SOAP クライアントのコマンドライン・オプションは `misima` 本体 (図 10・24 頁, 表 9・28 頁) とほぼ同じである。ただし、`-r rcfile` は SOAP クライアントではサポートしていない。またユーザ辞書機能は利用できない。

■ Java ランチャ起動 上記説明のラッパースクリプトを用いる方法ではなく、`java` クラスランチャで `misimaSoapClient.class` を直接コールする方法を示す。この場合環境変数 `CLASSPATH` に `misima.`



jar へのパスが含まれていなければならない。java の `-cp` オプションでクラスパスを指定してもよい。

この方法では `-u URI` で接続エンドポイントを必ず指定しなければならない。

次は UNIX での実行方法である。Windows も同様である。Windows での Java クラスランチャによる実行の様子は、図 21 (38 頁) の二つ目の実行例のとおりである。

```
% java misimaSoapClient -kyitq -s c \  
-u "http://host/axis/services/misimaSoapConnector"\  
< inputfile > outputfile
```

もし企業ネットワークのように、Squid などのプロキシを経由して外部インターネットワークと接続しているならば、上記方法は利用できない。次のようにプロパティによってプロキシ環境を指定すると利用できる場合がある。

```
% java \  
-Dhttp.proxyHost="プロキシホスト名" \  
-Dhttp.proxyPort="プロキシポート番号" \  
-Dhttp.proxyUser="ユーザ名" \  
-Dhttp.proxyPassword="パスワード" \  
misimaSoapClient -kyitq -s c \  
-u http://host/axis/services/misimaSoapConnector \  
< inputfile > outputfile
```

`misima.sh`、`misima.bat` を利用する場合は、このプロキシ設定をスクリプトに組み入れる。SOAP 用 Emacs をプロキシ経由で利用する際は、この変更は必須となる。

### 7.3.2 SOAP コマンドライン実行例

UTF-8 で作成したテキストファイルをコマンドラインからネットワークを介して misima SOAP WebService サーバで変換する例を示す。図 22 (42 頁) は、ローカル PC (Mac OS X Tiger) の端末コマンドライン (ターミナル) から筆者サイト上の SOAP WebService サーバ—URI: <http://yasuda.homeip.net/axis/services/misimaSoapConnector> (FreeBSD) にアクセスして変換する前後のスクリーン・ショットである。Windows クライアントでもコマンドプロンプトから同様の変換を行うことができる。

### 7.3.3 SOAP Emacs

Emacs からの misima SOAP WebService 実行は “`M-x misimasoap-コマンド RET`” とする。利用可能な Emacs コマンドを表 16 (43 頁) に示す。 `-region` 付きのコマンドは選択リージョンに対して変換を行う。

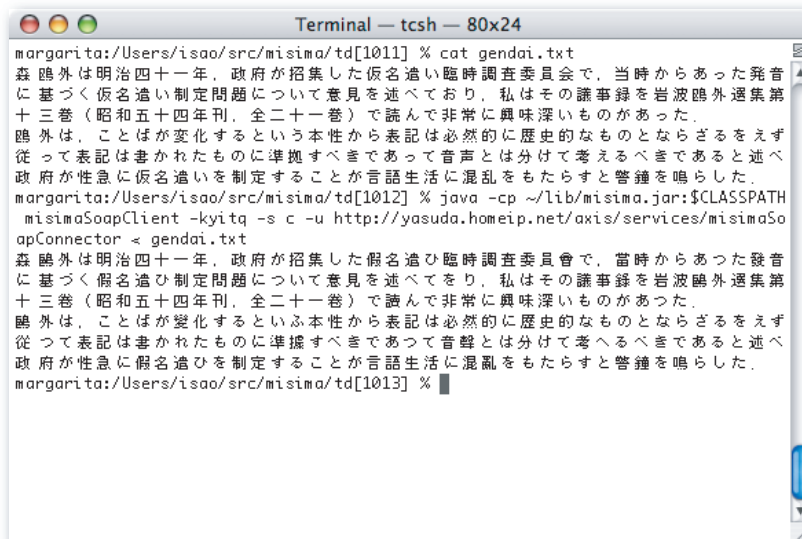


図 22 misima SOAP WebService (変換前後)

Windows XP Meadow 3.00 環境上での Emacs SOAP クライアントによる変換操作を図に示す。図 23 (42 頁) はリージョンを指定して変換コマンドを発行する直前を、図 24 (42 頁) は変換実行後を示している。

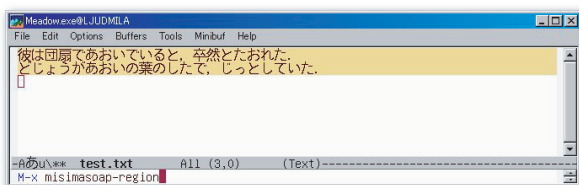


図 23 Windows Meadow 変換前

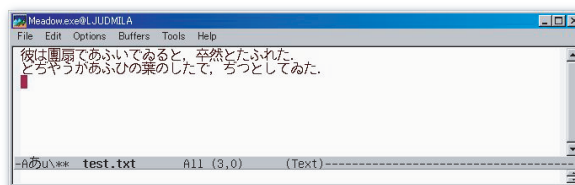


図 24 Windows Meadow 変換後

## 7.4 Microsoft Word SOAP クライアント

Microsoft Word クライアントはリモートサーバにある misima SOAP サーバに接続し、あたかも目の PC で変換しているような利用形態をサポートする。SOAP WebService は、二つの互いに独立したアプリケーションが必ずしも Web を前提に作られていなくても、ごく小さな SOAP コネクタを準備して比較的容易にこれらを連携させることができる。

本節では misima SOAP WebService Word クライアントのセットアップと使い方について説明する。パッケージ添付の Word クライアントは筆者サイトの misima SOAP サーバに固定接続するものである。

### 7.4.1 Microsoft Office XP Web Services Toolkit

Microsoft は Office 製品における Web サービスクライアント開発のためのツールキット Microsoft Office XP Web Services Toolkit 2.0 を無償で配布している。misima Word クライアントの実行にはこのツールキットが必要である。以下の手順でセットアップする。

表 16 SOAP Emacs コマンド一覧

コマンド	機能
<code>misimasoap-region</code>	<code>= -kyit -s c</code>
<code>misimasoap-tex-region</code>	<code>= -kyit -s a -x a</code>
<code>misimasoap-kyuji-region</code>	<code>= -t -s c</code>
<code>misimasoap-kyuji-tex-region</code>	<code>= -t -s a -x u</code>
<code>misimasoap-kyukana-kyuji-region</code>	<code>= -kt -s c</code>
<code>misimasoap-kyukana-kyuji-tex-region</code>	<code>= -kt -s c -x u</code>
<code>misimasoap-kanbun-region</code>	<code>= -s a -x k</code>
<code>misimasoap-invertkana-region</code>	<code>= -n</code>
<code>misimasoap-buffer</code>	<code>= -kyit -s c<sup>*1</sup></code>
<code>misimasoap-kyuji-buffer</code>	<code>= -t -s c<sup>*1</sup></code>
<code>misimasoap-tex-buffer</code>	<code>= -kyit -s a -x a<sup>*1</sup></code>
<code>misimasoap-kyuji-tex-buffer</code>	<code>= -t -s a -x u<sup>*1</sup></code>
<code>misimasoap-create-buffer</code>	<code>= -kyit -s c<sup>*2</sup></code>
<code>misimasoap-tex-create-buffer</code>	<code>= -kyit -s a -x a<sup>*2</sup></code>

\*1 当該バッファ

\*2 新規バッファ

1. Microsoft Office Web サイト <http://www.microsoft.com/japan/office/previous/xp/developer/webservices/download.msp> (図 25・44 頁) から Office XP Web Services Toolkit 2.0 をダウンロードする。
2. `setup.exe` を実行し、インストールを行う。
3. インストールが完了したら、Microsoft Word を起動して利用可能な状態にあるか確認する。「ツール (T)」-「マクロ (M)」-「Visual Basic Editor (V)」を順にクリックし (図 26・44 頁)、Visual Basic Editor を表示させる。次に「ツール (T)」-「Web Service References (W)」をクリック (図 27・44 頁) して、Web Service References Tool が表示されればよい (図 28・45 頁)。

#### 7.4.2 Word テンプレートの格納

misima Word クライアントの VBA (Visual Basic for Applications) マクロ及びメニューを備えた Word テンプレート `misimaTemplate.dot` をインストールする。Office マクロウイルスが蔓延している現在、企業 PC などではマクロテンプレートのインストールがセキュリティ上不可能になっている場合があるので、運用を確認して設定を行うこと。

パッケージ `¥soap¥tools¥misimaTemplate.dot` を Microsoft Word のスタートアップフォルダ：`C:¥Documents and Settings¥Username¥Application Data¥Microsoft¥Word¥STARTUP` にコピーする。ドライブ名や `Username` は利用者の環境に応じて読み替える。

これで、Microsoft Word を起動すると、misima クライアントが利用できる。

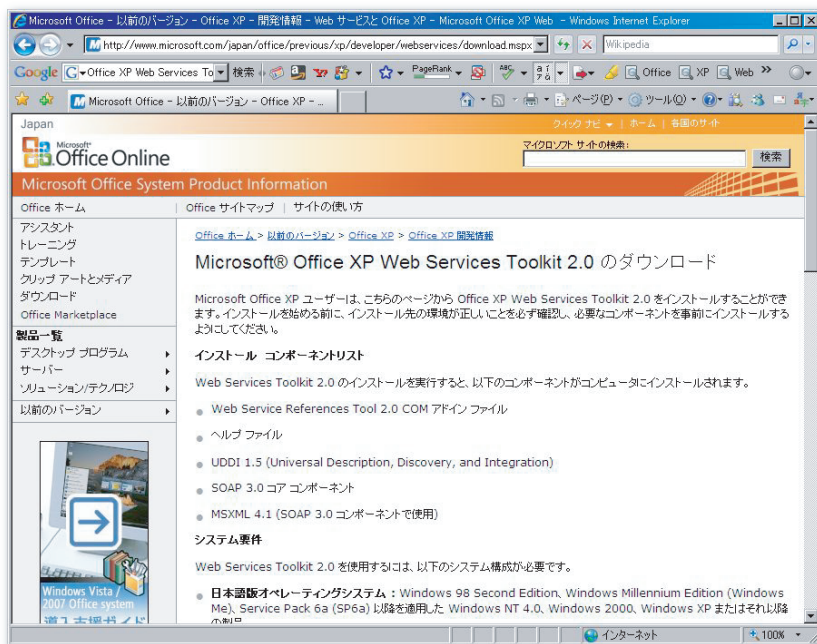


図 25 Web Services Toolkit ダウンロードサイト

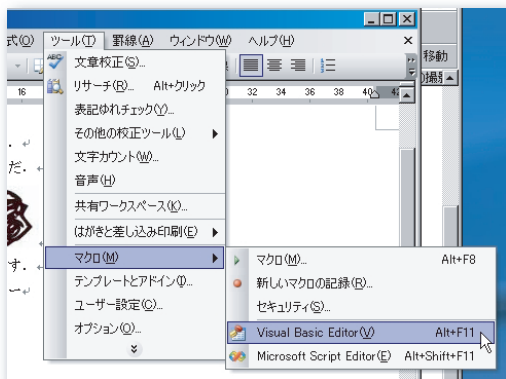


図 26 Visual Basic Editor

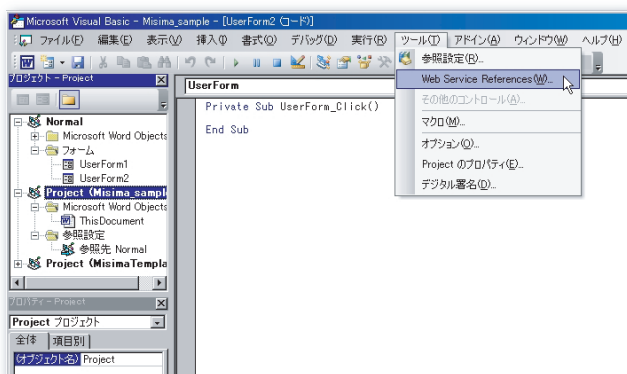


図 27 Web Service References Tool の表示

### 7.4.3 Word クライアントの使い方

`misimaTemplate.dot` には、予めツールバーと右クリックショートカットに変換操作のためのメニューが追加されている。

文書を作成しながら、必要なテキストを選択する。このとき必ずテキストのみの連続領域を選択しなければならない。ここでツールバーにある「旧仮名・旧字変換」をクリックすると変換処理がはじまる。その様子を図 29 (45 頁) に示す。

ネットワーク経由で筆者サイトの `misima` サーバに変換要求を行うため、時間がかかるかも知れない。変換が終了すると選択領域が変換結果で置換される (図 30・45 頁)。

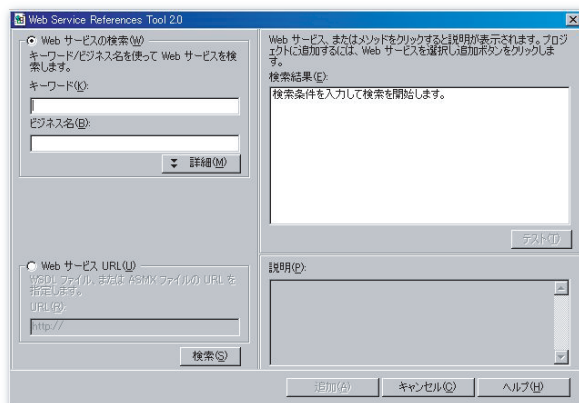


図 28 Web Services References Tool

ツールバー「旧仮名・旧字変換」ボタンは misima の `-kytiq -s c` オプション (表 9・28 頁) に相当するものである。これに対し、マウス右クリックショートカットメニューには、同じ変換に基づくオプション「旧仮名・旧字・用語・繰返変換」(`-kytiq -s c`) のみならず、「旧仮名・旧字変換」(`-ktq -s c`)、「旧字変換」(`-tq -s c`) が用意されている (図 31・46 頁)。

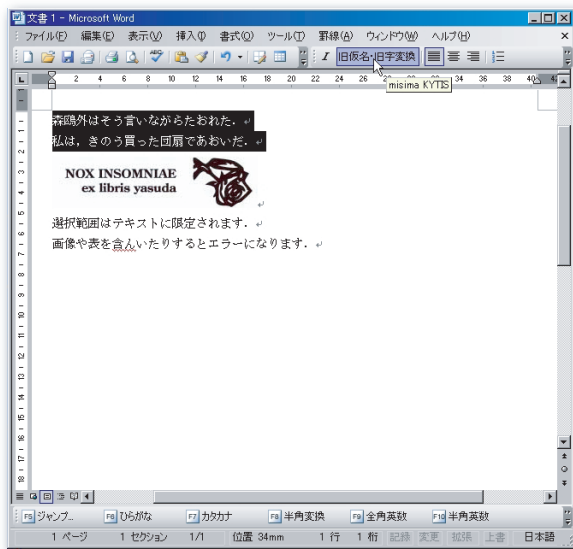


図 29 Word クライアント変換前

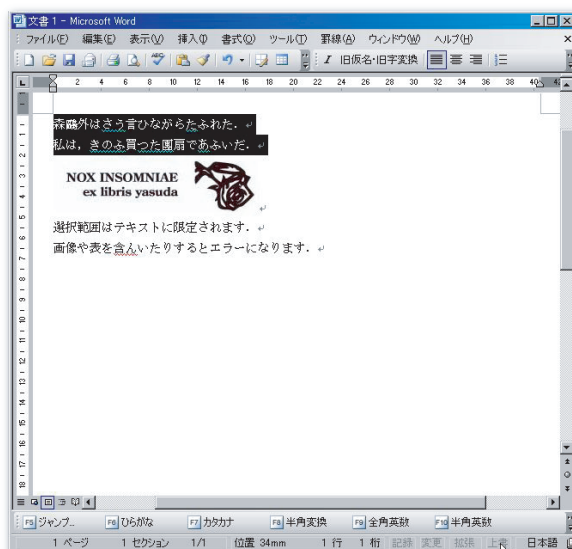


図 30 Word クライアント変換後

変換対象テキストの文字数は 1 回あたり 8,000 文字まで可能である。

提供テンプレートでは TeX 系の変換メニューは用意していない。Microsoft Word マクロの基礎知識があれば、「ツール (T)」-「マクロ (M)」から VBA マクロコードを呼び出して、独自にオプションを追加・修正すればよい。

Office XP Web Services Toolkit が生成した SOAP スタブクラスは、企業内のプロキシ環境下にある PC では、Internet Explorer のプロキシ設定を参照してプロキシ接続を行うようになっている。

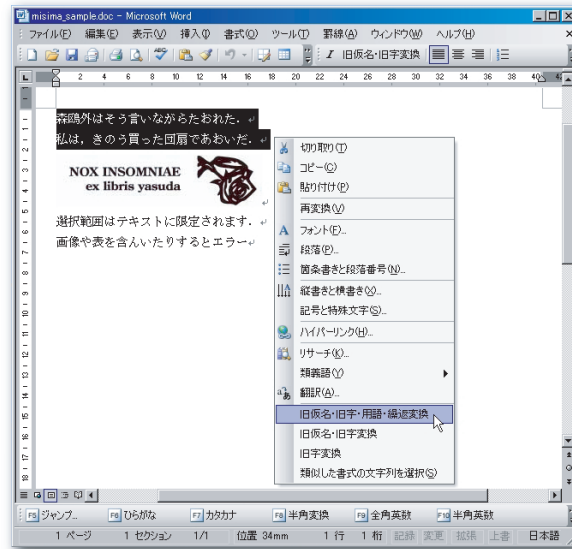


図 31 右クリックショートカットメニュー

変換操作において注意すべき点がある。変換対象の選択において、図 32 (46 頁) に示すように画像・表などのオブジェクトを含めてしまうと、SOAP スタブクラスの XML パーサでエラーとなる (図 33・46 頁)。必ず連続したテキスト領域だけを選択して misima 変換指示を行わなければならない。

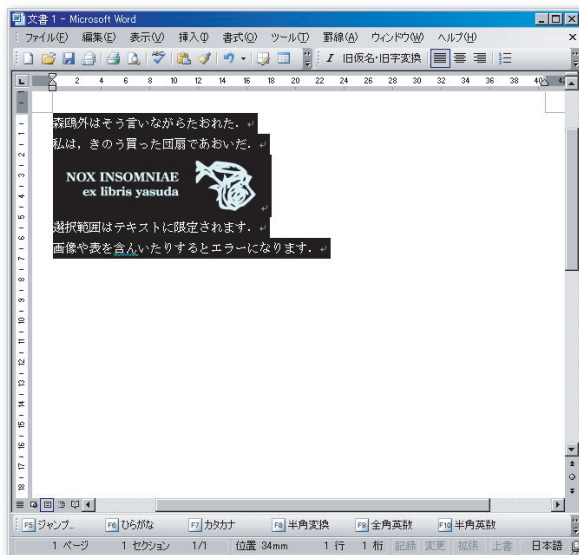


図 32 テキスト以外の選択は不可

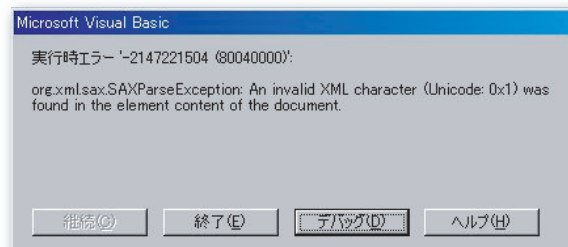


図 33 XML パーサエラー

#### 7.4.4 Microsoft Office クライアント追加について

Office XP Web Services Toolkit を用いると、Office VBA Web サービスクライアントの追加は極めて容易である。misima クライアントを追加する方法を簡単に説明する。

Web Service References Tool において、「Web サービス URL(U)-URL(R)」に misima SOAP サーバ WSDL アドレス <http://yasuda.homeip.net/axis/services/misimaSoapConnector?wsdl> を入力して「検索 (S)」を実行すると、図 34 (47 頁) のような検索結果が得られる。misimaSoapConnectorService にチェックを入れて「追加 (A)」をクリックすると、misima SOAP サービスのためのスタブクラスコードが自動生成される (35・47 頁)。

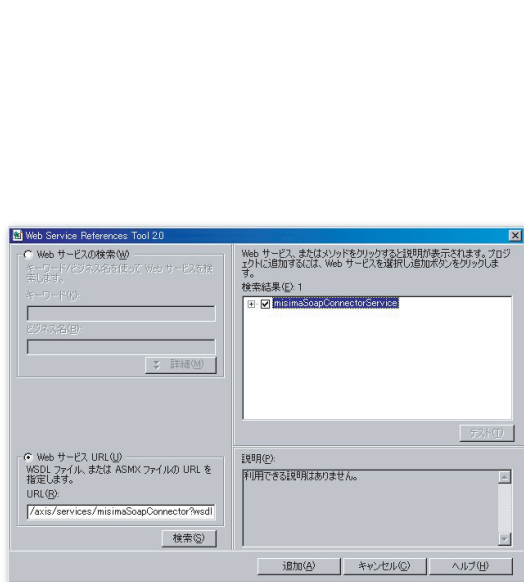


図 34 misima SOAP スタブクラスの追加

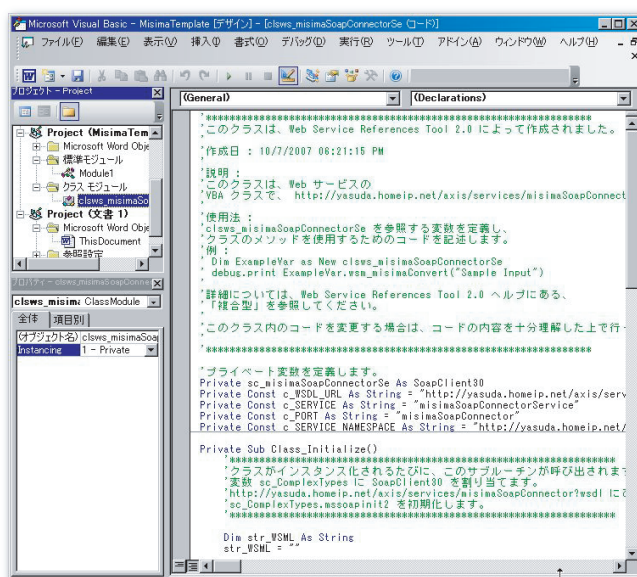


図 35 misima SOAP スタブクラスコード

この自動生成された `cls_misimaSoapConnectorSe` クラスからインスタンスを生成して、独自のクライアント VBA コードを書けばよい。misima 変換に必要なパラメータと対象テキストを指定して、`wsm_misimaConvert` メソッドを呼び出すだけである。エラー処理などを除いたその中核は次のようなコードで終わってしまう。

```
Sub misimaWebServiceConvert()
    ' 変換結果
    Dim sConvertResult As String
    ' 第一引数: misima 変換オプション
    Dim sArgument As String
    ' 第二引数: misima 変換対象テキスト (Unicode)
    Dim sTargetText As String
    ' SOAP 接続クラスの生成
```

```
Dim misimaWebService As New clsWS_misimaSoapConnectorSe
' misima 変換オプションのセット
sArgument = "-kyitq -s c"
' 選択範囲の取得
sTargetText = Selection.Text
' SOAP misima 変換サーバ呼び出し
sConvertResult = misimaWebService.wsm_misimaConvert( _
    sArgument, sTargetText)
' 変換結果で選択文字列を置換
Selection.Text = sConvertResult
End Sub
```

筆者サイトの misima SOAP サーバにアクセスするクライアントを利用者独自に作成・運用する場合は、筆者題記電子メールアドレスまで必ず連絡をいただきたい。



## 8 入力テキスト仕様

処理対象となるデータは Unicode UTF-8 でエンコードされた電子テキストとする。辞書などの周辺入力リソースについても同様である。

ロシア語等外国語は、茶釜では一文字ごとに分割されて、記号もしくは名詞の一種として扱われる。本プログラムでは、 $\text{T}_{\text{E}}\text{X}$  多言語文字変換指定がない限り、そのまま出力する。 $\text{T}_{\text{E}}\text{X}$  多言語文字変換指定がなされた場合も、変換テーブルに存在しないサポート対象外 UTF-8 文字はそのまま出力される。

空白文字（スペース、タブ）で区切られた文字列単位または行単位で処理を行う。よって文の途中でスペースや改行が入る場合、その位置によっては、茶釜の形態素解析が失敗して思った変換ができない可能性がある。とくに HTML などマークアップされたテキストを直接処理するときには注意が必要である。

形態素解析の関係上、連続する空白文字はスペース一個に置き換えて出力される。

## 9 misima 辞書・ $\text{T}_{\text{E}}\text{X}$ 変換テーブル

旧仮名遣い変換、用字・用語変換、旧字変換、単純変換は misima 辞書に、また  $\text{T}_{\text{E}}\text{X}$  多言語文字変換は  $\text{T}_{\text{E}}\text{X}$  変換テーブルに依存する。

misima 辞書は、筆者が準備した必須のシステム辞書と、利用者が独自に定義できるユーザ辞書から成る。ユーザ辞書を優先的に参照する。システム辞書もユーザ辞書もフォーマットは同一であり、定義できる内容にも差異はない。ユーザ辞書は任意であり、なくてもよい。

misima 辞書は旧仮名変換定義、旧字変換定義、用字・用語変換定義、単純変換定義から成る。

$\text{T}_{\text{E}}\text{X}$  変換テーブルは多言語変換テーブル (`euro-tbl`)、JIS テーブル (`jis-tbl`)、教会スラヴ語変換テーブル (`ocs-tbl`)、T2A キリル変換テーブル (`t2a-tbl`)、拡張変換テーブルから成る。このうち、拡張変換テーブルは所定のディレクトリ下に任意名称でユーザが追加できる。ユーザで指定できる  $\text{T}_{\text{E}}\text{X}$  変換定義には  $\text{T}_{\text{E}}\text{X}$  文字変換定義と  $\text{T}_{\text{E}}\text{X}$  異体字変換定義とがある。いずれも拡張変換テーブルとして作成する。

misima 辞書と  $\text{T}_{\text{E}}\text{X}$  変換テーブルとを同一テキストファイルに記述することはできない。

辞書のフォーマットの共通の書式は、先頭に辞書エントリ種別を示す固定文字と、空白文字で区切って定義内容とから構成される。

以下、辞書・テーブルにおける各種定義仕様について説明する。

### 9.1 旧字変換定義

新字とこれに対する旧字を定義する。新字-旧字の対応は一対一である。一新字に複数の旧字が対応する場合があるが、システム辞書では筆者の判断で特定の文字に割当てている。意味によって別の旧字を当てるべき場合、用字・用語変換定義 (9.3 節・52 頁) または単純変換定義 (9.4 節・53 頁) で熟語のパターンとして登録する。システム辞書にも「弁」などの文字について、熟語で判断可能な使い分けが登録されている。

変更したい場合はユーザ辞書で利用者が定義する。

システム辞書における旧字変換対象文字は文献 [4] を参考にし、さらに文献 [6] で妥当性を検証している。

悪名高い JIS83 年版で新字体に置き換えられてしまった文字（「鷗」などのいわゆる拡張新字体、丙種・包摂漢字）や Windows JIS 外字（「教」、「清」など）も適宜組み入れた。

システム辞書に登録されている全定義文字については、付録 A 新字-旧字対応表 (84 頁) を参照。

旧字変換の結果は、T<sub>E</sub>X OTF パッケージ CID 番号参照がもっとも充実している。UTF-8, JIS 第三・四水準にない異体字も登録している。「近」(二点付きしんじょうの「近」)、「唐」(縦線が「口」にめがけて突き出た「唐」) など、古風な異体字を筆者の判断で収録した。

旧字変換定義のフォーマットを図 36 (50 頁) に、各定義内容を表 17 (50 頁) に示す。

s 新字 旧字#FFFF#99999#999999

図 36 旧字変換定義

表 17 旧字変換定義内容

定義	説明
s	固定
新字	変換対象文字
旧字	変換後文字
#FFFF	Unicode UCS2 コード
#99999	CID 番号
#999999	今昔文字鏡大漢和番号

新字、旧字 はそれぞれ一文字のキャラクタである。# 以降は HTML 数値参照等のための情報で、記述は任意である。省略した場合、変換指定が HTML 数値参照であってもキャラクタ (旧字) に変換する。

第 1 エリア (#FFFF) は旧字として UCS2 コード (いわゆる Unicode コードポイント) を十六進数 4 桁で記述する。旧字変換における HTML 数値参照及び T<sub>E</sub>X OTF UCS 十六進数値参照で利用される。

第 2 エリア (#99999) は Adobe-Japan1.5 の CID 番号を十進数で指定する。T<sub>E</sub>X OTF CID 数値参照で利用される。

第 3 エリア (#999999) は今昔文字鏡番号を十進数で指定する。T<sub>E</sub>X 今昔文字鏡番号参照で利用される。

第 1~3 エリアにおいて 0 (#0) が指定された場合は、例えば T<sub>E</sub>X 今昔文字鏡番号変換指定を行ったとしても、0 参照 (\MO{0}) ではなくキャラクタ (旧字) に変換する。

s 悦 悦#6085#8455#010629

この定義例は以下を示す。

- 変換指定が UTF-8 文字 (-s c) のとき、悦 を 悦 に変換する。
- 変換指定が HTML 数値参照 (-s h) のとき、悦 を &#x6085 に変換する。

- 変換指定が T<sub>E</sub>X OTF UCS 十六進数値参照 (-s u) のとき、悦を `\UTF{6085}` に変換する。
- 変換指定が T<sub>E</sub>X OTF CID 数値参照 (-s a) のとき、悦を `\CID{8455}` に変換する。
- 変換指定が T<sub>E</sub>X 今昔文字鏡番号参照 (-s m) のとき、悦を `\MO{010629}` に変換する。

## 9.2 旧仮名変換定義

misima 辞書旧仮名変換定義は主に、本則変換 (11.2 節・60 頁) の対象外となる例外表記のデータベースである。

2005.5.13 辞書 Ver.1.0 では旧仮名定義は約 4,500 語が登録されている。これは、茶筌辞書 ipadic 2.7.0 の固有名詞以外の語から、本則の適用できないものをツールで抽出し、手作業で修正・登録したものである。旧仮名表記は文献 [5] で確認した。

システムロジックでは字音仮名遣いはとくに考慮していない。ただし、茶筌辞書に登録された平仮名表記の漢語で「わいうえお」(語中・語末) または「ずじ」が含まれるものは、筆者が妥当と思われる表記を選択的に定義している。例えば、「しょうとう」は「しやうたう (正当)」「しやうとう (正統, 掌灯)」「しやうとう (昇騰)」「しやうたう (松濤)」などがあるが、もっとも頻度が高いと思われる「せうとう (消灯)」に変換されるはずである。

旧仮名変換定義のフォーマットを図 37 (51 頁) に、各定義内容を表 18 (51 頁) に示す。

k 9 新仮名 旧仮名

図 37 旧仮名変換定義

表 18 旧仮名変換定義内容

定義	説明
k	固定
9	品詞コード
新仮名	変換対象語 (活用語は終止形)
旧仮名	変換後旧仮名表記 (活用語は終止形)

動詞、形容詞、助動詞については基本形 (終止形) で登録する。

対象テキスト (語・出現形) の基本形にマッチした **新仮名** を、旧仮名に置換する。以下「登録語」とは、**旧仮名** に指定している、変換後の単語を示す。活用する品詞 (動詞、形容詞、助動詞) はシステムで語幹部分を判断して置換する。ユーザ辞書で名詞、動詞、形容詞、副詞以外を定義することはあまりないと思う。

**9 新仮名** のうち **9** は品詞コードである。0~9 までの数字で指定する。品詞分類の考え方は茶筌に準ずる。

表 19 (52 頁) に品詞コードの一覧を示す。

旧仮名変換定義の例を次にあげる。順に名詞、動詞、形容詞の例である。

表 19 品詞コード

品詞コード	意味
0	名詞
1	動詞
2	形容詞
3	副詞
4	連体詞
5	接頭詞
6	接続詞
7	助詞
8	助動詞
9	感動詞その他

k 0 おとこ をとこ  
k 1 あおぐ あふぐ  
k 2 すえおそろしい すゑおそろしい

### 9.3 用字・用語変換定義

用字・用語変換定義は、旧仮名・旧字で書かれた古い文章に特徴的に現れる表記を個別に設定し、独立して変換できるようにするものである。定義に依存して、例えば「イングランド」を「英蘭」、「いわゆる」を「所謂」、「愛欲」を「愛慾」に置き換える。表記を古風にしたいときに使う。

用字・用語は時代、社会階層によって一律ではない。本パッケージの定義は文献 [4] を参考にした。用字・用語変換定義のフォーマットを図 38 (52 頁) に、各定義内容を表 20 (53 頁) に示す。

y 変換前 変換後

図 38 用字・用語変換定義

対象テキスト（語）の **変換前** を **変換後** に置換する。品詞に依存せず、また語幹の判断をせず置換するため、原則、活用しない品詞の単語を指定する。

旧字変換は用字・用語変換の後に独立して行うので、用字・用語変換定義の **変換後** 表記を必ずしも旧字で定義する必要はない。「辨別」「辯護」のように「弁」の旧字変換を区別したいときは旧字で定義しておく（こ

表 20 用字・用語変換定義内容

定義	説明
y	固定
変換前	変換対象語
変換後	変換後表記

の 2 語はシステム辞書に登録してある).

例を次にあげる.

y 画期的 劃期的  
y 兄嫁 嫂  
y エジプト 埃及

#### 9.4 単純変換定義

単純変換定義は、茶筌の解析がどうもうまく行かない、なにをおいても置き換えたい場合の補正のために設けている。語の統一にも利用できると思うが、もっぱら旧仮名遣い変換、旧字変換と同時に指定すべき補足機能であり、単独でオンにしても目的に合わないと考えたほうがよい。

処理の最後で繰返し符号変換を実施した後に、定義に従って差し換える。各種変換変換処理のシーケンスは 10.2 節 (58 頁) を参照。

マーキングの対象外である。

対象テキスト断片の **変換前** に完全一致する文字列を **変換後** に置換する。茶筌の語分割結果に依存せず置換する。

単純変換定義のフォーマットを図 39 (53 頁) に、各定義内容を表 21 (54 頁) に示す。

t 変換前 変換後

図 39 単純変換定義

例を以下にあげる.

t 缺伸 欠伸

表 21 単純変換定義内容

定義	説明
て	固定
変換前	変換対象語
変換後	変換後

旧字変換で一律「欠」は「缺」に変換されてしまう。これを補正し、本来の「欠伸」とする。

て 高島屋 高島屋

茶筌が「高島屋」を「高島」と「屋」に語分割しても「高島屋」に変換する。

て づ仕舞ひ ず仕舞ひ

旧仮名遣い変換で誤って「[やら]づ仕舞ひ」としたものを「[やら]ず仕舞ひ」に補正する。

て 々 ゝ

同ノ字点「々」を「ゝ」（ゆすり点）に変換する（これはシステム辞書には登録されていない）。

## 9.5 T<sub>E</sub>X 文字変換定義

T<sub>E</sub>X 多言語文字変換機能で利用する。UTF-8 文字に対して置換したい T<sub>E</sub>X 文字／コントロールシーケンスを指定する。

.misimarc の EXT\_DIR パラメータに指定した拡張変換テーブル・ディレクトリの下に、任意の名称でテーブルを追加できる。

マーキングの対象外である。

T<sub>E</sub>X 文字変換定義のフォーマットを図 40 (55 頁) に、各定義内容を表 22 (55 頁) に示す。

対象テキストの 変換前 の文字を 変換後 に置換する。以下の例は多言語変換テーブルに登録されているものから示している。

e	変換前	変換後
---	-----	-----

図 40 T<sub>E</sub>X 文字変換定義

表 22 T<sub>E</sub>X 文字変換定義内容

定義	説明
e	固定
変換前	変換対象文字
変換後	変換後 T <sub>E</sub> X シーケンス

e	Ж	ZH
---	---	----

キリル文字 Ж を“ZH”に変換する。

e	ḥ	>`h
---	---	-----

古典ギリシア文字 ḥ (無気氣息記号/重アクセント/下書きのイオタ付きエータ η) を“>`h|”に変換する。

e	€	\EURtm{}
---	---	----------

€ (ユーロ記号) を \EURtm{} に変換する。 \EUR{}、 \EURcr{} などにしたい場合はユーザ辞書として再定義いただきたい。

## 9.6 T<sub>E</sub>X 異体字変換定義

「▼」を前置した文字を、登録した異体字 (CID 数値参照形式) で出力することができる。T<sub>E</sub>X 異体字変換定義は独自に異体字を登録するためのものである。

マーキングの対象外である。

T<sub>E</sub>X 異体字変換定義のフォーマットを図 41 (56 頁) に、各定義内容を表 23 (56 頁) に示す。

c 変換前 変換後

図 41 T<sub>E</sub>X 異体字変換定義

表 23 T<sub>E</sub>X 異体字変換定義内容

定義	説明
c	固定
変換前	変換対象文字
変換後	変換後 T <sub>E</sub> X シーケンス

対象テキストの **変換前** の文字が ▼ の直後に指定されたとき、▼**変換前** を **変換後** に置換する。この場合、▼は削除される。仮に本異体字文字変換定義に登録されていない文字に対し ▼ が前置されていた場合、▼は通常の文字として扱い、削除されない。

システム辞書多言語変換テーブルに登録されている異体字を表 24 (57 頁) に示す。

## 9.7 misima 辞書登録時の注意事項

旧仮名変換及び用字・用語変換における misima 辞書検索は、茶筌によって分割された語単位で行う。ipadic (茶筌辞書) にない語は、単語として切り出されず、思わぬ語分割がされることがあり、よって misima 辞書への登録が無意味であることに注意すべきである。このような場合、語分割に依存しない単純変換定義を活用する。本システムでは、旧仮名遣い変換に際して必要と思われる語彙を茶筌辞書にも追加登録してチューニングしている。

misima は同一バッファのテキストに対し数種の変換を一定のシーケンス (10.2 節・58 頁) に従って行う。そのつど変換を受ける過程で、辞書に登録しても結果的に無意味となる可能性がある。例えば、仮に旧仮名遣い変換で「あい」が「ある」に変換された後では、単純変換定義で「あい ⇒ あひ」と登録しても「あひ」が得られることは永劫にない。各種変換有無を選択できることが事情をさらに複雑にしている。システム辞書ではこのチューニングを行っているが、ユーザ辞書を追加する際は注意すべきである。今後のシステムの課題であると考えられる。



表 24 T<sub>E</sub>X 異体字登録文字

入力	出力	組版結果
▼吉	\CID{13706}	吉
▼高	\CID{8705}	高
▼崎	\CID{14290}	崎
▼碕	\CID{17009}	碕
▼浜	\CID{8531}	濱
▼濱	\CID{8531}	濱
▼角	\CID{13682}	角
▼第	\CID{13910}	才
▼間	\CID{13693}	間
▼芦	\CID{7961}	芦
▼淫	\CID{4580}	姪
▼窓	\CID{5932}	窗
▼連	\CID{13517}	聯
▼群	羣	羣
▼翻	翻	翻
▼餅	餅	餅
▼略	畧	畧
▼隣	鄰	鄰
▼欲	慾	慾
▼糾	紕	紕
▼萌	萌	萌

## 10 処理方式概念

### 10.1 日本語形態素解析

あらゆる変換処理に先立って、misima は対象テキストを日本語形態素解析ソフトウェアである茶筌 Version 2.3.3 に委ねる。茶筌は日本語テキストに対し、辞書 ipadic Version 2.7.0 に基づいて形態素解析を行い、単語に分割し、出現語の基本形／品詞／活用情報を misima に返却する。misima は原則として切り出された単語情報ごとに変換操作を行う。

### 10.2 変換処理シーケンス

各種変換処理は以下の順番で行う。ユーザが機能選択をオフにした場合、当該処理はスキップする。

1. T<sub>E</sub>X 漢文訓点変換
2. 用字・用語変換
3. 旧仮名変換
4. T<sub>E</sub>X くノ字点変換
5. 旧字変換
6. 繰返し符号変換
7. 単純変換
8. T<sub>E</sub>X 多言語文字変換
9. 仮名反転変換

## 11 旧仮名変換仕様

旧仮名変換のプロセスは次のとおりである。

1. 日本語形態素解析で得られた語を misima 辞書で検索する。
2. misima 辞書に登録されている場合、辞書変換処理を行う。
3. 登録されていない場合、本則変換処理を行う。
4. 品詞依存処理を行う。
5. 拗音・促音変換処理を行う。

### 11.1 辞書変換処理

茶筌解析情報から品詞コード + 基本形で misima 辞書旧仮名変換定義テーブルを検索する。ヒットしたとき、表 25 (59 頁) のとおり変換する。ヒットしないとき、本則変換を行う。

なお、基本形とは活用する品詞では終止形と等価であり、名詞など活用しないケースでは出現形に等しい。登録語とは、旧仮名変換定義の **変換後** で指定した旧仮名表記語を指す。

表 25 旧仮名辞書変換

品詞	変換仕様	例
動詞	出現形が基本形と一致しないとき語幹部分を登録語の語幹部分で置換する.	出現形：植え [て]; 基本形：植える; 登録語：植ゑる; ⇒ 植え [て] を 植ゑ [て] に置換する.
形容詞／助動詞	出現形が基本形と一致しないとき登録語末尾「い」前で置換する.	出現形：こうごうしかっ [た]; 基本形：こうごうしい; 登録語：かうがうしい; ⇒ こうごうしかっ [た] を かうがうしかっ [た] に置換する.
その他	出現形を登録語で置換する.	不めいりょう [な] を 不めいれう [な] に置換する.

## 11.2 本則変換処理

旧仮名遣いの規則（「本則」と呼ぶ）を適用して変換する工程である。

本則は原則的に、福田恆存著『私の國語教室』（文献 [3]）での基準を適用する。「わいうえお」「じ」「ず」を「はひふへほ」「ぢ」「づ」に置換する、あるいは置換しない規則である。

表 26（62 頁）に本則変換仕様を示す。

説明のなかで用いている「動詞 五段・カ行イ音便」等の用語は茶筌及び ipadic の分類である。詳細は文献 [1, 2] を参照のこと。

## 11.3 品詞依存処理

辞書変換、本則変換に係りなく、品詞解析結果に基づき、表 27（63 頁）変換を行う。

## 11.4 拗音・促音変換処理

拗音・促音「やゆよつやユヨツ」を「やゆよつやユヨツ」に置換する。ただし片仮名のみからなる名詞は対象としない。

## 12 用字・用語変換仕様

茶筌解析結果の語基本形で misima 辞書用字・用語変換定義テーブルを検索する。ヒットしたとき登録語（変換後表記）で置換する。ヒットしないときなにもしない。

変換操作は単純に登録語で置き換えるだけであり、活用を考慮しない。従って、ユーザ辞書に登録する際は、名詞など活用しない品詞の語のみとすべきである。

## 13 単純変換仕様

対象テキスト行単位でテキスト断片が misima 辞書単純変換定義に一致するとき登録テキスト断片で置換する。

旧仮名遣い変換、用字・用語変換では茶筌解析結果の語の単位で処理を行うのに対し、単純変換では語境界に依存せず置換する。

## 14 繰返し符号変換仕様

以下の基準に従う。

- 半濁音（ぱぴふぺぽパピプペポ）は変換対象外である。
- 語頭に濁音をもつ場合（「ぢぢ」）の繰返し部は「>」（「ぢ>」）とする。仮に一語内に「ぢぢ」を有する場合も「ぢ>」となる。
- 文献 [7] に準拠し、原則として語の境界を越えて適用はしない。しかしながら、文献 [7] は定義としてはあまりにも曖昧であり計算機処理上明確に判断基準とできる性質のものではない。その背景は、そも

そも繰返し符号の用例が実表記において統一性を欠いているからにほかならない。議論の余地はあると思うが、本システムでは以下の例外を採用している。

- 一語でも固有名詞及び、日本語として一般化した片仮名のみからなる名詞（「ママ」など）には適用しない。
- 語の境界を越えて適用する例外パターンは以下のとおり。
  - \* 動詞連用形 + 「て」 [助詞-接続助詞] : 「立て + て」パターン（「立て > 」）
  - \* 名詞（-地域） + 名詞（-地域） : 多く「～町 + 町内」パターン（「内幸町々内」）
  - \* 「～と」 [助詞-格助詞]（する） : 「こと + とする」パターン（「すること > する」）「～などという」の「と」 [助詞-格助詞-引用] は適用外。
  - \* 「～で」 [助詞] + 「でも」 [接続詞] : 「～で + でも」パターン（「言葉で > も」）「でも」 [助詞]，「で」 [助詞] へと続く場合は適用外。
  - \* 「～の」 [助詞-連体化] + 名詞 : 「をの（斧） + の刃」パターン（「をの > 刃」）「をのの壊れるのは悲しい」の最初の「の」 [助詞-格助詞] は適用外。

## 15 TeX 変換仕様

### 15.1 多言語文字変換

多言語文字変換機能は、標準変換、拡張変換、中韓変換、タイ語変換、キリル T2A 変換、十六進数変換、強制変換からなる。変換仕様は、強制変換機能及び制御タグ仕様を除き、拙作 `Utf82TeX`<sup>\*23</sup> と同一である。

#### 15.1.1 標準変換

ラテン文字言語、スラヴ語（ロシア語、ウクライナ語、ブルガリア語などの現代スラヴ語、古スラヴ語、教会スラヴ語）、ギリシア語（古典・現代）、日本語異体字、中国語簡体字、中国語繁体字、韓国語の文字を、`pTeX` で扱える文字もしくは命令（コントロールシーケンス）に変換する。Unicode 多言語文字、Unicode 漢字を用いて `TeX` 文書を直接記述し、本機能で変換した後に JIS, SJIS もしくは EUC のテキストとして保存すれば、`pTeX` でコンパイル可能となる。<sup>\*24</sup> `Babel` パッケージ、`OTF` パッケージの利用を想定している。

標準変換は `-x u` オプションで機能が働く。

#### ラテン文字言語

Latin-1 (ISO 8859-1), Latin-2 (ISO 8859-2) の文字変換に対応している。T1 エンコーディングに準拠している。ISO 8859-1 で符号化可能な言語は、アイスランド語、アイルランド語、イタリア語、英語、オランダ語、スウェーデン語、スペイン語、デンマーク語、ドイツ語、ノルウェー語、ポルトガル語、フィンランド語、フェロー語、フランス語である。一方 ISO 8859-2 は、アルバニア語、英語、スロバキア語、セルビア・クロアチア語<sup>\*25</sup>、チェコ語、ドイツ語、ハンガリア語、ポーランド語、ルーマニア語をカバーしている。<sup>\*26</sup>

<sup>\*23</sup> 「UTF-8 による TeX 文書の作成」<http://yasuda.homeip.net/tex/utf82tex.html> を参照。

<sup>\*24</sup> 土村氏の `ptetex3`, 角藤先生の `W32TeX` の場合、`platex` オプションに `--kanji=utf8` を指定すれば、JIS, SJIS, EUC へのコード変換は不要である。

<sup>\*25</sup> 1992 年のユーゴスラビア解体によって名称は消滅。

<sup>\*26</sup> 文献 [10, pp.327-8] を参照。

表 26 旧仮名本則変換

文字	変換仕様
わ	<ul style="list-style-type: none"> <li>(1) 語中・語末では「は」に置換する。語頭では置換しない。</li> <li>(2) 語中・語末でも「わ」とすべきものは辞書に登録されるものとする：あわてる、いわし、etc.</li> </ul>
い	<ul style="list-style-type: none"> <li>(1) 語中・語末では「ひ」に置換する。語頭では置換しない。</li> <li>(2) 形容詞 のとき語末の「い」は置換しない。</li> <li>(3) 動詞 五段・カ行イ音便 のとき語末の「い」は置換しない：「突い (動詞 五段・カ行イ音便) -て」</li> <li>(4) 動詞 五段・ガ行 のとき語末の「い」は置換しない：「急い (動詞 五段・ガ行) -で」</li> <li>(5) 動詞 五段・ラ行特殊 のとき「い」は置換しない：「ください (動詞 五段・ラ行特殊 連用) -ます」</li> <li>(6) 助動詞／形容詞・イ段 のとき「い」は置換しない。</li> <li>(7) 助動詞 特殊・ナイ のとき「い」は置換しない。</li> <li>(8) 助動詞 特殊・タイ のとき「い」は置換しない。</li> <li>(9) その他語中・語末で「い」とすべきものは辞書に登録されるものとする：さいはひ、etc.</li> <li>(10) 「ゐ」に変換すべき語は辞書に登録されるものとする。</li> </ul>
う	<ul style="list-style-type: none"> <li>(1) 語中・語末では「ふ」に置換する。語頭では置換しない。</li> <li>(2) 動詞ウ音便／形容詞 のとき語末の「う」は置換しない：「問う (動詞 五段・ワ行ウ音便) -て」</li> <li>(3) 名詞-非自立-助動詞語幹「よう [な]」の「う」は置換せず「やう [な]」とする。</li> <li>(4) 名詞-接尾-助動詞語幹「そう [な]」の「う」は置換せず「さう [な]」とする。</li> <li>(5) その他語中・語末で「う」とすべきものは辞書に登録されるものとする：はうき、かうじ、etc.</li> </ul>
え	<ul style="list-style-type: none"> <li>(1) 語中・語末では「へ」に置換する。語頭では置換しない。</li> <li>(2) 語中・語末でも「え」とすべきものは辞書に登録されるものとする：おぼえ、もえぎ、etc.</li> <li>(3) 「ゑ」に変換すべき語は辞書に登録されるものとする。</li> </ul>
お	<ul style="list-style-type: none"> <li>(1) 語中・語末では「ほ」に置換する。語頭では置換しない。</li> <li>(2) 語中・語末でも「お」とすべきものは辞書に登録されるものとする：はおり、物おぢ、etc.</li> <li>(3) 「を」に変換すべき語は辞書に登録されるものとする。</li> </ul>
じ	<ul style="list-style-type: none"> <li>(1) 「ぢ」に置換する。</li> <li>(2) 「感じる (一段)」など基本形で「じる」、基本形以外語末「じ」のとき、「じる」「じ」を「ずる」にして茶筌形態素解析を行い、サ変・ズル 基本型 で返される場合は「ぢ」としない：信じる、甘んじる、etc.</li> <li>(3) 「すさまじい (形容詞・イ段)」型の「じ」は「ぢ」としない。</li> <li>(4) 上記以外で「じ」とすべき語は辞書に登録されるものとする。</li> </ul>
ず	<ul style="list-style-type: none"> <li>(1) 「づ」に置換する。</li> <li>(2) 「感ずる (サ変・ズル)」の「ず」は「づ」としない</li> <li>(3) 助動詞 特殊・ヌ の「ず」は「づ」としない。</li> <li>(4) 上記以外で「ず」とすべき語は辞書に登録されるものとする。「負けず嫌い」など一語で定着している (正確には茶筌辞書に一語として登録されている) 名詞の「ず」は、助動詞 特殊・ヌ の「ず」に由来するものとは判断できないため辞書に登録すべきである。</li> </ul>
のう	<ul style="list-style-type: none"> <li>(1) 「せつろう [ございます]」などの「のう」を「なう」に置換する。これは茶筌の誤解析への対応であり、「の」(助詞-連体化) 「う」(感動詞) が連続するとき、ナイ形容詞 ウ音便 と同等と判断する。</li> </ul>
とう	<ul style="list-style-type: none"> <li>(1) 「行きとう [ございます]」などの「とう」を「たう」に置換する。これは茶筌の誤解析への対応であり、動詞連用形に「とう」(副詞-助詞類接続) が連続するとき行う。</li> </ul>

表 27 品詞依存処理

(助) 動詞 未然ウ接続 または 助動詞 特殊・ダ 未然 (だろ [う]) のとき、末尾の「お」列文字を「あ」列文字に置換する。ただし「や」行の場合を除く。

例	言おう	⇒ 言ほう	⇒ 言はう
	行こう	⇒ 行かう	
	だろう	⇒ だらう	
	見よう	⇒ 見よう	(置換しない)
	しよう	⇒ しよう	(置換しない)

形容詞 連用ゴザイ接続 のとき、「しゅう」「じゅう」をそれぞれ「しう」「じう」に置換する。

例	美しゅう [ございます]	⇒ 美しう
	すさまじゅう [ございます]	⇒ すさまじう

助動詞 特殊・デス 未然形 「でしょ [う]」、特殊・マス 未然ウ接続 「ましよ [う]」をそれぞれ「でせ [う]」「ませ [う]」に置換する。

### キリル文字言語

Unicode Cyrillic U+0400–U+04FF<sup>\*27</sup> をカバーしている。現代ロシア語、ウクライナ語、ブルガリア語のみならずアゼルバイジャン語、バシキール語、タジク語、古スラヴ語、教会スラヴ語などの文字変換が可能である。OT2 エンコーディング (ASCII 翻字対応) で表すことができる文字は OT2 優先で変換する。それ以外の文字は T2A, T2B, T2C, T2D エンコーディングに変換する。

### キリル T2A 変換

`<misima_t2a>` タグと `</misima_t2a>` タグの間のキリル文字を、キリル T2A 拡張オプション指定 (15.1.2 節・65 頁) によらず T2A シンボルに変換する。`<misima_t2a>`, `</misima_t2a>` タグは出力されない。

### 教会スラヴ語 $\LaTeX$ 記法変換

A. Слепухин 氏が開発した教会スラヴ語  $\LaTeX$  パッケージで採用されている教会スラヴ語表記を OldSlav で処理できる命令に変換する。`<misima_ocs>` タグと `</misima_ocs>` タグの間にテキストを記述する。`<misima_ocs>`, `</misima_ocs>` タグは出力されない。一方 `<misima_ocs>`, `</misima_ocs>` タグ範囲内には教会スラヴ語・古スラヴ語文字は、`euro-tbl` テーブルに従って T2D エンコーディングで変換する。表 28 (64 頁) で両者を比較せよ。

表 28 教会スラヴ語変換比較

記法	記述	出力	$\TeX$ 出力
$\LaTeX$ 記法	<code>&lt;misima_ocs&gt;А&lt;/misima_ocs&gt;</code>	<code>\OCSYUSM{}</code>	А
T2D	А	<code>\CYRLYUS{}</code>	А

### ギリシア語

Unicode Greek U+0370–U+03FF<sup>\*28</sup> (ただしコプト語は除く) 及び Greek Extended U+1F00–U+1FFF<sup>\*29</sup> の範囲を対象とする。現代ギリシア語のほか複式アクセントを有する古典ギリシア語をサポートする。基本、Babel LGR エンコーディングに変換する。一部 Teubner パッケージの命令に準拠したものもある。

### ハングル

Unicode ハングルと判断できるコード範囲 (U+AC00–U+D7A3, U+3130–U+318F, U+1100–U+11FF) の文字は `\UTFK{USC2 コード}` に変換する。

### Unicode 漢字

JIS X 0208 範囲外の Unicode CJK 統合漢字を `\UTFM{USC2 コード}` に変換する。`\UTFM` 命令は日本語にグリフがなく繁体字、簡体字、ハングルにあればそのグリフを選択してくれる。

### 中韓文字

`<misima_cjk_x>` タグ `x` 指定値に応じて、CJK 統合漢字を中韓 OTF シーケンスとして出力する。`</misima_cjk>` が現れると元の変換基準に復帰する。`x:c` のとき `\UTFC{USC2 コード}` (中国語

<sup>\*27</sup> <http://www.unicode.org/charts/PDF/U0400.pdf>

<sup>\*28</sup> <http://www.unicode.org/charts/PDF/U0370.pdf>

<sup>\*29</sup> <http://www.unicode.org/charts/PDF/U1F00.pdf>



簡体字), `x:t` のとき `\UTFT{UCS2 コード}` (中国語繁體字), `x:k` のとき `\UTFK{UCS2 コード}` (韓国語ハンチャ) に変換する. Unicode CJK 統合漢字は日本語/中国語/韓国語で微妙に字体の異なる表記を同じコードに帰してしまう (包摂) が, OTF シーケンスを用いることにより各国語の字体を表現できる. `<misima_cjk_x>`, `</misima_cjk>` タグは出力されない. `\UTFM`, `\UTFK`, `\UTFC`, `\UTFT` は OTF パッケージ `multi` オプション指定時に利用できるコントロールシーケンスである. CJK 統合漢字を一律中国語簡体字として扱いたいような場合は中韓変換オプション (15.1.2 節・65 頁) を用いる.

#### 特殊異体字

吉, 高などの文字に「▼」を前置すると, それぞれ「吉 (上部が土の吉)」, 「高 (梯子高)」を出力する CID 数値参照命令 `\CID{13706}`, `\CID{8705}` に変換する. 登録文字については表 24 (57 頁) を参照.

$\TeX$  のフォント選択命令, 言語環境指定命令 (`\selectlanguage{russian}`, `\fontencoding` 等) はユーザで指定する必要がある. 変換対象はあくまでも文字に限定される.

$\TeX$  多言語変換オプションを選択すると, ユーザが旧字変換において「UTF-8 文字出力」または「HTML 数値参照」を指定したとしても「 $\TeX$  OTF CID 数値参照」で旧字変換を行う.

### 15.1.2 拡張/中韓/タイ語/キリル T2A /十六進数/強制変換

#### 拡張変換

`.misimarc EXT_DIR` ディレクトリ下にある拡張変換テーブルを用いて変換を行う. 標準変換の対象文字と拡張変換テーブルの定義が重複する場合, 後者を優先する. 複数の拡張変換テーブル間で変換対象文字の定義が重複する場合, Perl `readdir` 関数が返す拡張変換テーブルファイル名の順番に依存して, 後から読み込まれた定義を優先する. 提供パッケージはベトナム語, グルジア語テキストをそれぞれ  $\text{Vn}\TeX$ , `mxedruli` パッケージで処理できる形式に変換するテーブル, 機種依存文字<sup>\*30</sup>テーブル (「@」「Ⅶ」「Ꞇ」「𐌀」「𐌁」などが処理できる) を含んでいる. ユーザが独自に追加したテーブルで変換する場合, 本オプションを使う. [`-x x` オプション]

#### 中韓変換

Unicode CJK 統合漢字をオペランドに応じて, **C** オペランド: `\UTFC` (中国語簡体字), **T** オペランド: `\UTFT` (中国語繁體字), **K** オペランド: `\UTFK` (韓国語) の OTF シーケンスに変換する. これを指定すると, 原則仮名以外のすべての漢字に適用される. [`-C C|T|K` オプション]

#### タイ語変換

タイ語を  $\text{p}\TeX$  で処理するために, Unicode タイ語テキストを TIS-620 コードに変換し, タイ語単語境界 `{\wbr}` を挿入し, さらに「`^^十六進数`」形式に変換する. U+0E00–U+0E7F 範囲の Unicode 文字をタイ文字と判断する. 「`^^十六進数`」に変換するのは日本語以外の 8 ビットコードではエラーとなる  $\text{p}\TeX$  でもこの形式だと受け付けるためである. [`-x t` オプション]

#### タイ語変換 (UTF-8 出力)

タイ語変換結果を「`^^十六進数`」形式ではなく Unicode タイ文字で出力する. [`-x T` オプション]

<sup>\*30</sup> ここでいうところの機種依存文字とは表示・印刷がオペレーティングシステム処理系に依存する文字を示している. しかしながら `misima` はあらゆる機種依存の文字をサポートしているわけではなく, Windows の機種依存文字から拾ったものであり, テーブルに登録された Unicode 文字に限る. 変換対象文字は `contrib/sjis-tbl` を参照.

### キリル T2A 変換

キリル文字を可能な限り T2A コントロールシーケンスで出力する。この指定がないと OT2 エンコーディング (ASCII 翻字形式) で出力する。[-x r オプション]

### 十六進数変換

サポート対象外の (変換テーブルにない) UTF-8 文字を「^^十六進数」形式に変換する。[-x h オプション]

### 強制変換

無変換指定のなされた (<misima\_noop>, </misima\_noop> タグで囲まれた) 範囲であっても、変換対象とする。[-x f オプション]

## 15.1.3 T<sub>E</sub>X 変換例

本節では T<sub>E</sub>X 多言語変換の適用例を示す。misima がサポートする言語を直接タイプして T<sub>E</sub>X 原稿を UTF-8 で準備し、これを misima により変換し、pL<sub>A</sub>T<sub>E</sub>X 2<sub>ε</sub> でタイプセットしたものである。コマンドライン手順は図 42 (66 頁) のとおりである。なお、例のタイプセットのために T<sub>E</sub>X 原稿のプリアンプルには、図 43 (67 頁) に示す内容を指定した。

```
% misima -kyitq -s a -x uxt < ex.tex > ex.utf8
% platex --kanji=utf8 ex.utf8
% dvipdfmx ex.dvi
```

図 42 misima 変換, pL<sub>A</sub>T<sub>E</sub>X タイプセット手順

ラテン文字言語について、図 44 (68 頁) に misima への入力を、図 45 (69 頁) に misima 変換結果で組版した出力を示す。

キリル文字言語例を入力：図 46 (70 頁)，組版結果：図 47 (71 頁) に示す。

古典ギリシア語例を入力：図 48 (71 頁)，組版結果：図 49 (72 頁) に示す。

タイ語例を入力：図 50 (72 頁)，組版結果：図 51 (72 頁) に示す。

タイ語単語境界には `swath` によって `{\wbr}` が挿入される。ここで適切に改行されるようにするには、次のようにコントロールシーケンスの定義をしておく必要がある。

```
\def\wbr{\hskip0pt plus0.6pt minus0.6pt\relax}%
```

ヴェトナム語例を入力：図 52 (73 頁)，組版結果：図 53 (73 頁) に示す。

CJK 統合漢字 (日本語／中国語／韓国語) 例を入力：図 54 (74 頁)，組版結果：図 55 (75 頁) に示す。

```

\documentclass[a4paper]{jsarticle}
\usepackage{inputenc}%
\usepackage[10pt]{typelec}%
\usepackage[OT2,T2A,T2B,T2C,T2D,T1]{fontenc}%
\usepackage[%
    french,%           フランス語
    german,%          ドイツ語
    czech,%           チェコ語
    russian,%         ロシア語
    ukrainian,%       ウクライナ語
    oldchurchslavonic,% OldSlav 教会スラヴ語
    polutonikogreek,% 古典ギリシア語
    thai,%            ThaiLaTeX タイ語
    vietnam,%         VnTeX-3 ヴェトナム語
    nippon%           日本語 (筆者自作: OldSlav 添付)
]{babel}%           Babel 多言語
\usepackage{thswitch}% ThaiLaTeX encoding switch
\usepackage{dblacnt}% VnTeX3
\usepackage{vietnam}% VnTeX-3
\usepackage{sfkanbun}% 漢文訓点パッケージ (藤田先生)
\usepackage{teubner}% トイブナー古典語
\usepackage{marvosym}% ユーロ記号等
\usepackage{textcomp}% 記号等
\usepackage{psycyr}%   PSCyr キリル Type1
\usepackage[deluxe,expert,multi]{otf}% OTF 和文 (齋藤氏)

```

図 43 文書例プリアンブル

```
\selectlanguage{french}
\section{フランス語 Français}
Souvent, la main portée au front, debout sur les vaisseaux, tandis que la lune se balançait
entre les mâts d'une façon irrégulière, je me suis surpris, faisant abstraction de tout ce
qui n'était pas le but que je poursuivais, m'efforçant de résoudre ce difficile problème!

\hfill {\em Conte de Lautréamont, «Les Chants de Maldoror»}%

\hfill\today

\selectlanguage{german}
\section{ドイツ語 Deutsch}
Dann ließ sie ihre Hände langsam über meine Wangen heruntergleiten, und ihr Blick ruhte mit
unendlicher Innigkeit auf mir. Sie schüttelte den Kopf mit einem schmerzlichen Ausdruck,
als könnte sie irgend etwas nicht fassen. „Mußst du denn schon heute weg?“ fragte sie
leise.

\hfill {\em A. Schnitzler, »Die Frau des Weisen«}%

\hfill\today

\selectlanguage{czech}%
\section{チェコ語 Czech}
Posláním sdružení je vytvářet předpoklady pro všestranné využívání a další rozvoj jazyka
počítačové typografie  $\TeX$  a příbuzného programového vybavení pro stolní tisk, zejména
mezi českými a slovenskými uživateli.

\hfill {\em Czechoslovak  $\TeX$  Users Group}%

\hfill\today
```

図 44 ラテン文字言語入力例

### 佛蘭西語 Français

Souvent, la main portée au front, debout sur les vaisseaux, tandis que la lune se balançait entre les mâts d'une façon irrégulière, je me suis surpris, faisant abstraction de tout ce qui n'était pas le but que je poursuivais, m'efforçant de résoudre ce difficile problème!

*Conte de Lautréamont, «Les Chants de Maldoror»*

7 octobre 2007

### 獨逸語 Deutsch

Dann ließ sie ihre Hände langsam über meine Wangen heruntergleiten, und ihr Blick ruhte mit unendlicher Innigkeit auf mir. Sie schüttelte den Kopf mit einem schmerzlichen Ausdruck, als könnte sie irgend etwas nicht fassen. „Mußt du denn schon heute weg?“ fragte sie leise.

*A. Schnitzler, »Die Frau des Weisen«*

7. Oktober 2007

### チェコ語 Czech

Posláním sdružení je vytvářet předpoklady pro všestranné využívání a další rozvoj jazyka počítačové typografie  $\text{\TeX}$  a příbuzného programového vybavení pro stolní tisk, zejména mezi českými a slovenskými uživateli.

*Czechoslovak  $\text{\TeX}$  Users Group*

7. října 2007

図 45 ラテン文字言語組版出力例

```

\selectlanguage{russian}
\section{ロシア語 %
\protect\fontencoding{OT2}\protect\selectfont{Русский}
\fontencoding{OT2}\selectfont%
Прежде всего откроем тайну которую Мастер не пожелал открыть Иванушке. Возлюбленную его
звали Маргаритой Николаевной. Все, что Мастер говорил о ней, было сущей правдой. Он описал
свою возлюбленную верно. Она была красива и умна.

\hfill
{\em M. Bulgakov, <Master i Margarita>}\par
\hfill\today

\selectlanguage{ukrainian}%
%%<misima_t2a> t2a control
\section{ウクライナ語 %
\protect\fontencoding{T2A}\protect\selectfont Українська}
\fontencoding{T2A}\selectfont%
\textac{% PSCyr Akademicheskaja font
Відвідуючи меморіальні музеї Гете, Пушкіна, Толстого, кожен звертає увагу на колосальні
особисті бібліотеки цих титанів світової культури. Має і український народ подібного
велера. Це---Тарас Шевченко. }
%%</misima_t2a>

\hfill
{\em \textlatin{\tt http://kobzar.info/our/library/}}\par
\hfill\today

\selectlanguage{nippon}%
\section{教会スラヴ語\protect\SlavTeX{}}
{\selectlanguage{oldchurchslavonic}%
%%<misima_ocs> OldChurchSlavonic control
''Аще же под\ъ кл'ятвоу <ил'и длуч'єніємъ <архієр'єйскимъ <ил'и <ієр'єйскимъ б'ысть, <ил'и
''аще кл'ятву <отц'а своег'w <ил'и м'атере свое'я навед'є на с'я, <ил'и своем'у прокл'ятію
подпад'є, <ил'и кл'ятву преступ'и, <ил'и <ин'ыми н'экіими грэх'и ''јкв челов'экъ связ'ася:
%%</misima_ocs>

\hfill
{\em \SlavTeX{} \textlatin{sample}}\par
\hfill\today}

```

図 46 キリル文字言語入力例

### 露西亞語 Русский

Прежде всего откроем тайну которую Мастер не пожелал открыть Иванушке. Возлюбленную его звали Маргаритою Николаевной. Все, что Мастер говорил о ней, было сущей правдой. Он описал свою возлюбленную верно. Она была красива и умна.

*М. Булгаков, «Мастер и Маргарита»*  
7 октября 2007 г.

### 烏克蘭語 Українська

Відвідуючи меморіальні музеї Гете, Пушкіна, Толстого, кожен звертає увагу на колосальні особисті бібліотеки цих титанів світової культури. Має і український народ подібного велета. Це—Тарас Шевченко.

<http://kobzar.info/our/library/>  
7 жовтня 2007 р.

### 教會スラヴ語 ҃лѣТѢХ

Їще же по́дъ клѣтвомъ ѿлѣченїемъ архїерейскимъ ѿнѣ ѿрѣскимъ бысть, ѿнѣ ѿще клѣтвѣ ѿтца своегѡ ѿнѣ матерѣ своєѡ навєдѣ на сѡ, ѿнѣ своємѣ проклѣтїю подпадѣ, ѿнѣ клѣтвѣ престѣпнѣ, ѿнѣ ѿнымн нѣкїимн грѣхнѣ ꙗкѡ человекѣкѣ свѣзѣа:

҃лѣТѢХ sample  
7 октября 2007 г.

図 47 キリル文字言語組版出力例

```
\section{古典ギリシア語}
{\selectlanguage{polutonikogreek}}
\begin{verse}
~\textit{Ἄνδρα μοι ἔννεπε, Μοῦσα, πολύτροπον, ὃς μάλα πολλὰ\\
πλάγχθη, ἐπεὶ Τροίης ἱερὸν πτολίεθρον ἔπερσε.\\
πολλῶν δ' ἀνθρώπων ἴδεν ἄστεα καὶ νόον ἔγνω,\\
πολλὰ δ' ὄ γ' ἐν πόντῳ πάθεν ἄλγεα ὄν κατὰ θυμόν,\\
ἀρνύμενος ἥν τε ψυχὴν καὶ νόστον ἐταίρων.\\
ἀλλ' οὐδ' ὧς ἐτάρους ἐρρύσατο, ἰέμενός περ·\\
αὐτῶν γάρ σφετέρῃσιν ἀτασθαλίῃσιν ὄλοντο,\\
νήπιοι, οἳ κατὰ βοῦς Ἴπερίονος Ἥελίοιο\\
ἦσθιον· αὐτὰρ ὁ τοῖσιν ἀφείλετο νόστιμον ἦμαρ.\\
τῶν ἀμόθεν γε, θεά, θύγατερ Διός, εἰπέ καὶ ἡμῖν.}
\end{verse}
\hfill [ \textit{~Ομηρος} ] \makebox[10em]{}
```

図 48 古典ギリシア語入力例

### 古典希臘語

Ἄνδρα μοι ἔννεπε, Μοῦσα, πολύτροπον, ὃς μάλα πολλὰ  
 πλάγχθη, ἐπεὶ Τροίης ἱερὸν πτολίεθρον ἔπερσε.  
 πολλῶν δ' ἀνθρώπων ἴδεν ἄστεα καὶ νόον ἔγνω,  
 πολλὰ δ' ὃ γ' ἐν πόντῳ πάθεν ἄλγεα ὃν κατὰ θυμόν,  
 ἀρνύμενος ἣν τε ψυχήν καὶ νόστον ἐτάϊρων.  
 ἀλλ' οὐδ' ὧς ἐτάρους ἐρρύσατο, ἰέμενός περ·  
 αὐτῶν γὰρ σφετέρῃσιν ἀτασθαλίῃσιν ὄλοντο,  
 νήπιοι, οἳ κατὰ βοῦς Ἰπερίονος Ἡελίοιο  
 ἦσθιον· αὐτὰρ ὁ τοῖσιν ἀφείλετο νόστιμον ἦμαρ.  
 τῶν ἀμόθεν γε, θεά, θύγατερ Διός, εἰπέ καὶ ἡμῖν.

[ Ὅμηρος ]

图 49 ギリシア語出力例

```
\selectlanguage{thai}%
\section{タイ語 ภาษาไทย}
\def\wbr{\hskip0pt plus0.6pt minus0.6pt\relax}%
\fontencoding{LTH}\fontfamily{norasi}\selectfont%
เป็นมนุษย์สุดประเสริฐเลิศคุณค่ากว่าบรรดาฝูงสัตว์เดรัจฉานจงฝ่าฟันพัฒนาวิชาการอย่าล้างผลาญถูเช่นฆ่าปีทา
ใครไม่ถือโทษโกรธแข่งชดชืดชืดด่าหัดอภัยเหมือนกีฬาอัชฌาสัยปฏิบัติประพฤติกฏกำหนดใจพุดจาให้จ๊ะๆ จ๋าๆ นำฟัง
เอ๋ยๆ
\hfill\today
```

图 50 タイ語入力例

### タイ語 ภาษาไทย

เป็นมนุษย์สุดประเสริฐเลิศคุณค่ากว่าบรรดาฝูงสัตว์เดรัจฉานจงฝ่าฟันพัฒนาวิชาการอย่าล้างผลาญถูเช่นฆ่าปีทา  
 ใครไม่ถือโทษโกรธแข่งชดชืดชืดด่าหัดอภัยเหมือนกีฬาอัชฌาสัยปฏิบัติประพฤติกฏกำหนดใจพุดจาให้จ๊ะๆ จ๋าๆ นำฟัง  
 เอ๋ยๆ

7 ตุลาคม พ.ศ. 2550

图 51 タイ語組版出力例



```
\selectlanguage{vietnam}%
\section{ヴェトナム語 %
{\protect\fontencoding{T5}\protect\selectfont Tiếng Việt}}
\fontencoding{T5}\selectfont%
Phần “Những câu hỏi và giải đáp thường gặp”(viết tắt từ tiếng Anh là FAQ) được nêu ra ở đây
nhằm mục đích thu thập những câu hỏi thường gặp trong thực tế và những lời giải đáp thích
hợp nhất của nó. Từ lần ấn bản cuối cùng đến nay, đã có những sự thay đổi khá lớn trong cấu
trúc của FAQ. \textbf{Để hiểu rõ hơn bạn nên xem lại chương “Làm sao tôi có thể đọc được
FAQ”.}

\hfill{\em CJK samples}}\par
\hfill\today
```

図 52 ヴェトナム語入力例

## ヴェトナム語 Tiếng Việt

Phần “Những câu hỏi và giải đáp thường gặp” (viết tắt từ tiếng Anh là FAQ) được nêu ra ở đây nhằm mục đích thu thập những câu hỏi thường gặp trong thực tế và những lời giải đáp thích hợp nhất của nó. Từ lần ấn bản cuối cùng đến nay, đã có những sự thay đổi khá lớn trong cấu trúc của FAQ. **Để hiểu rõ hơn bạn nên xem lại chương “Làm sao tôi có thể đọc được FAQ”.**

*CJK samples*

Ngày 7 thángmườinăm 2007

図 53 ヴェトナム語組版出力例

```

\section{日本語}
雪後庵は起伏の多い小石川界隈の高台にあって、幸いに戦災を免かれた。
三千坪に及ぶ名高い小堀遠洲流の名園と共に、京都のとある名刹から移された中雀門も、奈良の古い寺をそのまま
移した玄関や客殿も、あとに建てられた大広間も、何一つ損なわれていなかった。

戦後の財産税さわぎの只中に、雪後庵は元の持主の実業家の茶人の手から、美しい元気な女主人の手に渡つて、たち
まち名高い料理屋になった。

\vspace{1zh}%
\hfill 三島由紀夫『宴のあと』

\vspace{1zh}%
骨練平。

%%<misima_cjk_k>
\section{韓國語}
이 FAQ 은 자주 반복되는 질문과 그에 대한 대답을 간단명료한 양식으로 모아 엮어졌습니다.
이 FAQ 의 구조는 지난 판에 비하여 획기적으로 변경되었습니다.
상세한 것은 “이 FAQ 을 어떻게 읽을 것인가” 라는 대목을 참조하시기 바랍니다.

\vspace{1zh}%
骨練平。
%%</misima_cjk>

%%<misima_cjk_c>
\section{简体中文}
本常問問答集 (FAQ list)
是从一些经常被问到的问题及其适当的解答中、以方便的形式摘要而出的。
跟上一版不同的是、其编排结构已彻底改变。
有关新结构的细节、可参考如何阅读本问答集及了解其编排结构该项中的说明。

\vspace{1zh}%
骨练平。
%%</utf82tex>

%%<utf82tex_t>
\section{繁體中文}
本常問問答集 (FAQ list)
是從一些經常被問到的問題及其適當的解答中、以方便的形式摘要而出的。
跟上一版不同的是、其編排結構已徹底改變。
有關新結構的細節、可參考如何閱讀本問答集及了解其編排結構該項中的說明。

\vspace{1zh}%
骨練平。
%%</misima_cjk>

```

图 54 CJK 統合漢字入力例

## 日本語

雪後庵は起伏の多い小石川界限の高臺にあつて、幸ひに戦災を免かれた。三千坪に及ぶ名高い小堀遠洲流の名園と共に、京都のとある名利から移された中雀門も、奈良の古い寺をそのまま移した玄関や客殿も、あとに建てられた大廣間も、何一つ損なはれてゐなかつた。

戦後の財産税さわぎの只中に、雪後庵は元の持主の實業家の茶人の手から、美しい元氣な女主人の手に渡つて、たちまち名高い料理屋になつた。

三島由紀夫『宴のあと』

骨練平。

## 韓國語

이 FAQ 은 자주 반복되는 질문과 그에 대한 대답을 간단명료한 양식으로 모아 엮어졌습니다. 이 FAQ 의 구조는 지난 판에 비하여 획기적으로 변경되었습니다. 상세한 것은 “이 FAQ 을 어떻게 읽을 것인가” 라는 대목을 참조하시기 바랍니다.

骨練平。

## 简体中文

本常问问答集 (FAQ list) 是从一些经常被问到的问题及其适当的解答中、以方便的形式摘要而出的。跟上一版不同的是, 其编排结构已彻底改变。有关新结构的细节、可参考如何阅读本问答集及了解其编排结构该项中的说明。

骨练平。

## 繁體中文

本常問問答集 (FAQ list) 是從一些經常被問到的問題及其適當的解答中、以方便的形式摘要而出的。跟上一版不同的是、其編排結構已徹底改變。有關新結構的細節、可參考如何閱讀本問答集及了解其編排結構該項中的說明。

骨練平。

图 55 CJK 統合漢字組版出力例

## 15.2 漢文訓点変換

-x k オプションを指定すると、漢文訓点変換が機能する。変換対象となるテキスト要素を `<misima_kanbun>` と `</misima_kanbun>` で囲んで変換範囲を限定しなければならない。藤田眞作氏が公開している T<sub>E</sub>X 漢文訓点組版スタイル `sfkanbun.sty` に準拠したフォーマットに、バタの文字列を整形する。pT<sub>E</sub>X 縦組で使用する。

### 15.2.1 書き方の概念

misima における漢文訓点の書き方は、訓点付き漢文の組み上がりイメージにおいて次に示す順序要素からなるまとまり（グループ）を繰返し記述して行く。

1. 親文字 [漢字]
2. 振仮名 (読み) [平仮名] (省略可)
3. 送仮名 [片仮名] (省略可)
4. 返点, 音符号 [15.2.2 節・76 頁参照] (省略可)
5. 再読文字送仮名 [片仮名] (省略可)
6. 句読点 [、,。] (省略可)

送仮名や返点の漢文訓点表示のシーケンス・イメージをそのまま書いていけばよいのではないかと思う。sfkanbun.sty を使い慣れた方には逆に面倒かも知れないが、グループの指定シーケンスを図 56 (76 頁) に示す。

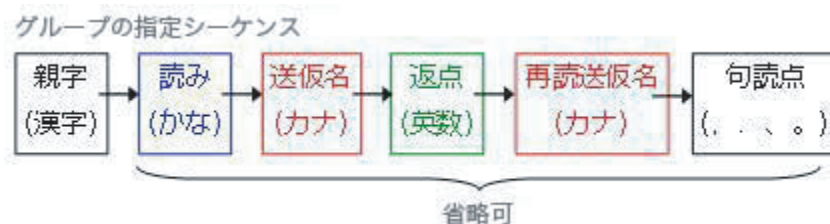


図 56 グループ指定シーケンス

変換例を図 57 (77 頁) に示す。また、変換におけるグループの概念を図 58 (77 頁) に示す。

### 15.2.2 グループ要素仕様

グループの各要素の仕様を示す。

- 対象テキストに対し漢文訓点変換を行うためには、T<sub>E</sub>X 漢文訓点変換オプション `-x k` を指定した上、さらに変換したいテキスト範囲を、`<misima_kanbun>` 開始タグと `</misima_kanbun>` 終了タグで囲む必要がある。このタグは制御用であり、T<sub>E</sub>X 漢文訓点変換を指定した場合、処理結果には出力されない。T<sub>E</sub>X 変換指定のない場合はタグをそのまま出力する。それ以外は `<( / )misima_kanbun>` タグ中の「\_」を「\\_」に変えて出力する (T<sub>E</sub>X で処理することを考慮したため)。
- 振仮名、送仮名、返点 のいずれかが指定されない場合 (これは仮名などの文字種で判断している) は {

子曰ハク、盍なんゾ 3 ルト各々言ハ 2 爾なんぢノ志ヲ 1.



```
子\kundoku{日}{ }{ }{ハク}{ }{(、)}\kundoku{盍}{なん}{ゾ}{三}<ルト>各\ninojiten
\kundoku{言}{ }{ }{ハ}{二}\kundoku{爾}{なんぢ}{ノ}{ }\kundoku{志}{ }{ }{ヲ}{一}{ }{(。)}%
```

図 57 漢文訓点変換例

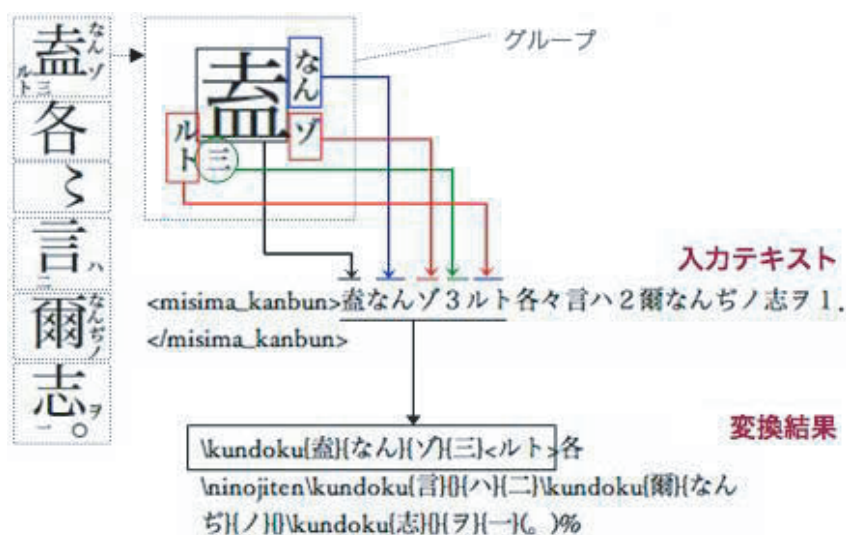


図 58 漢文訓点変換イメージ

} 内は空で整形する。いずれも指定されない場合はフォーマット変換せず **親文字** のみをそのまま出力する。従って **親文字** と **句読点** のみが連続する間は **親文字 句読点** のみ出力される。

- **再読文字送仮名**, **句読点** が指定されない場合、括弧 < >, ( ) も含め出力しない。
- **親文字** には括弧 (「」『』【】) が含まれる。
- **親文字** に続いて片仮名がくると **送仮名** と判断する。**再読文字送仮名** のみを指定することはできない。そもそも **再読文字送仮名** は再読時左側に送るものであり、漢字に続いて **再読文字送仮名** のみが指定されることはないはずである。
- **返点** は、1 (一) ~ 5 (五), J (上) C (中) G (下), K (甲) O (乙) H (丙), T (天) I (地) N (人), R (レ) のように英数字 (半角全角, 大文字小文字いずれも可) で指定する。変換後は ( ) 内に置き換わる。「1R」, 「JR」と指定するとそれぞれ \itireten (一とレの接合したもの), \uereten (上とレの接合したもの) に変換する。
- **返点** に単独でもしくは上記返点に続けて「-(ハイフン)」, 「-(全角ハイフン)」または「-(長音)」を記述すると、**音号符** に変換する。**音号符** とは、漢文訓読文のいわゆる熟語の字間に置く「-」のような訓

点である。「R」(レ点)とは混在できない。単独で「-」が指定されると `\ongofuno` に、「2-」,「G-」と指定されるとそれぞれ `\ongofuni`, `\ongofuge` に変換する。

- 「各々」における「々」は `\ninojiten` に変換する。
- 句読点は「...」が記述できる。「. . .」が指定された場合「\。」に変換する。
- 行末に % を出力する。
- `<misima_kanbun>`, `</misima_kanbun>` タグ範囲内では旧仮名変換処理, 用字・用語変換, 単純変換, 繰返し符号変換, 仮名反転処理は行わない。漢字を旧字体で出力したいなら, 旧字変換機能を併用する。

### 15.2.3 漢文訓点組版例

`misima` 記法による入力例とその変換結果組版出力とをそれぞれ図 59 (79 頁), 図 60 (80 頁) に示す。

## 15.3 くノ字点変換

TeX OTF パッケージでは縦組和文において「くノ字点濁点無 (`\ajKunoji`)」及び「くノ字点濁点付 (`\ajDKunoji`)」を出力する命令がサポートされている。本オプションは入力テキスト中の「そろそろ」「それぞれ」などの繰返し表現を自動的に「`そろ\ajKunoji{}`」「`それ\ajDKunoji{}`」などに変換する。

変換仕様を以下に示す。

- 変換対象は, その繰返し部の文字列が二文字以上からなり, かつ平仮名ないし片仮名を一文字以上含む, 繰返し表現である。
- 茶筌解析の結果で同一語の範囲内で変換を行う。例えば「`つるつるかめかめ`」は, 「`つるつる`」という表現は茶筌辞書に語として存在し変換対象となるのに対し, 「`かめかめ`」は「`かめ`」(亀) - 「`かめ`」(囃め)と二語で解析されるため変換されない。結果は「`つる\ajKunoji{}` `かめかめ`」となる。
- 同一文字が連続するもの(「`ハハハハハ`」等)は変換対象としない。別の繰返し符号変換機能によって「`ハ\ \ \ \`」とすることができる。
- 原則として, 変換対象は必ず平仮名, 片仮名が含まれる繰返し表現である。漢字のみのものは対象としない。当然ながら英数字・記号文字列も変換対象とはならない。「`不承不承`」は例外である。
- 先頭に漢字がくるものは繰返し部の読みの清濁が判断できないため, 原則「くノ字点濁点無」に変換する。「`散り散り`」など繰返し部で濁る表現は「くノ字点濁点付」とすべきであるが, 茶筌辞書に登録されている語(「`重ね重ね`」「`返す返す`」「`不承不承`」「`代る代る`」)及び, 「`散り散り`」「`冴え冴え`」「`懲り懲り`」「`沁み沁み`」など筆者の思い当たるいくつかは正しく変換されるようになっている。これ以外は手修正するか単純変換定義などで救うしかない。
- 本変換処理は, 別に支援する繰返し符号変換処理に先立って実行される。よって両機能をともに指定した場合, 「`ただただ`」は「`たゞ\ajKunoji{}`」となる。

くの字点変換による組版例を図 61 (80 頁) に示す。

```

\parindent=0pt%
\DeclareAJKundokuStyle%
\let\ninojiten=\ajvarNinoji%

%<misima_kanbun>
顔淵・季路待ス。子曰ハク、盍なんゾ3ルト各々言ハ2爾なんぢノ志ヲ1。
%</misima_kanbun>

\DeclareOriginalKundokuStyle%
\vspace{1.5zh}%
%<misima_kanbun>
楽シム G 得テ2天下之英才ヲ1 而教 C-育スルヲ之ヲ J。
%</misima_kanbun>

\vspace{1.5zh}%
%<misima_kanbun>
聞ク H 吏部ノ有リ G 以テ2 博学宏くわう辞じヲ1 選ブ者 J,
人尤もっとモ謂いヒ2之ヲオト1, 得うト O 美仕ヲ K。
%</misima_kanbun>

\vspace{1.5zh}%
%<misima_kanbun>
既ニ出デテ得2 其船ヲ1, 便チ扶そヒ2 向さきノ路みちニ1, 処処ニ誌しス R 之ヲ。
及ビ2 郡下ニ1, 詣いたりテ2 太守ニ1, 説クコト如シ R 此かくノ。
太守即すなはチ遣しムルモ G 人ヲシテ随したがヒテ2 其ノ往クニ1,
尋ネ C 向ニ所ヲ JR 誌セシ, 遂ニ迷ヒテ不2 復得1 R 路ヲ。
%</misima_kanbun>

\vspace{1.5zh}%
%<misima_kanbun>
闔かふ廬りよ曰ハク、「可キ3 試ミニ以テス2 婦人ヲ1 乎」。
曰ハク、「可ナリ」。
於テ R 是ここニ許ス R 之ヲ。
%</misima_kanbun>

```

図 59 漢文訓点変換例 (入力)

## 16 変換精度の制約・限界

本プログラムの変換精度は、計算機による機械的解析に基づく以上、制約、限界がある。ある程度の分量の文章に適用すると、変換誤りが含まれるものと思ったほうがよい。重要な文書に対してお利用になる場合には、必ず自分の目で変換結果をご確認いただきたい。以下、本システムの変換方式において本質的な問題を要因ごとに示す。

才<sup>ト</sup>、得<sup>ウ</sup>美<sup>ト</sup>仕<sup>甲</sup>。  
 聞<sup>ク</sup>吏<sup>ノ</sup>部<sup>リ</sup>有<sup>テ</sup>以<sup>テ</sup>博<sup>ク</sup>學<sup>ク</sup>宏<sup>ク</sup>辭<sup>ヲ</sup>選<sup>ブ</sup>者<sup>上</sup>、人<sup>モ</sup>尤<sup>ト</sup>謂<sup>ヒ</sup>之<sup>ヲ</sup>。  
 樂<sup>シ</sup>得<sup>テ</sup>天<sup>ノ</sup>下<sup>ヲ</sup>之<sup>ヲ</sup>英<sup>才</sup>而<sup>シテ</sup>教<sup>育</sup>之<sup>上</sup>。  
 顏<sup>ノ</sup>淵<sup>ノ</sup>・季<sup>ノ</sup>路<sup>ノ</sup>待<sup>ス</sup>。子<sup>ハ</sup>曰<sup>ク</sup>、盍<sup>ニ</sup>各<sup>々</sup>言<sup>フ</sup>爾<sup>ノ</sup>志<sup>ヲ</sup>。  
 既<sup>ニ</sup>出<sup>テ</sup>得<sup>テ</sup>其<sup>ノ</sup>船<sup>ヲ</sup>、便<sup>チ</sup>扶<sup>ヒ</sup>向<sup>キ</sup>路<sup>ヲ</sup>處<sup>ニ</sup>處<sup>ニ</sup>誌<sup>シ</sup>之<sup>ヲ</sup>。及<sup>ビ</sup>  
 郡<sup>ノ</sup>下<sup>ニ</sup>詣<sup>リ</sup>、太<sup>ノ</sup>守<sup>ノ</sup>、說<sup>ク</sup>如<sup>ク</sup>此<sup>ノ</sup>。太<sup>ノ</sup>守<sup>ノ</sup>即<sup>チ</sup>遣<sup>ハ</sup>下<sup>ニ</sup>  
 人<sup>ヲ</sup>隨<sup>ヒ</sup>其<sup>ノ</sup>往<sup>ク</sup>、尋<sup>ニ</sup>向<sup>キ</sup>所<sup>ヲ</sup>誌<sup>シ</sup>、遂<sup>ニ</sup>迷<sup>ヒ</sup>不<sup>レ</sup>復<sup>ス</sup>  
 得<sup>レ</sup>路<sup>ヲ</sup>。  
 闔<sup>カ</sup>廬<sup>ノ</sup>曰<sup>ク</sup>、「可<sup>キ</sup>試<sup>ミ</sup>以<sup>テ</sup>婦<sup>ノ</sup>人<sup>一</sup>乎<sup>ヲ</sup>」。曰<sup>ク</sup>、「可<sup>ナ</sup>リ」。  
 於<sup>テ</sup>是<sup>ニ</sup>許<sup>シ</sup>之<sup>ヲ</sup>。

図 60 漢文訓点変換例 (出力)

そろそろ  
 それぞれ  
 散り散り  
 そろく  
 それぐ  
 散りぐ

図 61 くの子点変換

## 16.1 辞書要因

辞書に定義されていないパターンはうまく置換できない場合がある。変換結果を評価してユーザで追加いただきたい。とりわけ動詞が名詞化した語に弱い。「負けず嫌い」などは一語として扱われるため、「ず」は否定の「ぬ」に由来することが茶筌の解析では判断できず、本則に従って「づ」に置換されてしまう場合がある(「負けず嫌い」そのものは辞書に登録してある)。

## 16.2 字音仮名遣いの問題

本システムは字音仮名遣いの正確な変換ができない。漢字の音読みについて、現代仮名遣いで「しょう」と書かれても、旧仮名では「しろう(松)」なのか「しやう(声)」なのか「せう(小)」なのかはたまた「せふ(妾)」なのか、判断できないことが多い。ある程度漢字仮名混じりの和文でないと、人間が見ても難しいものがある。ましてや機械は追従できない。辞書で一律置き換えることは可能であるが、意図しない結果を招く副作用も考慮すべきである。茶筌辞書に平仮名で登録された漢語の一部を misima 辞書に定義しているが、筆者の判断で表記を特定しているので、意図と合致するか利用者にて結果を確認する必要がある。いずれにせよ、漢語は漢字で入力するのが基本と考えていただきたい。

## 16.3 形態素解析要因

旧仮名変換解析の前提となっている茶筌そのものが形態素解析に失敗するために、適切に変換されない場合がある。これは対象の日本語が曖昧か、不適切であるか、もしくは語義の特定において文脈に著しく依存するケースが多い。ちょっとした誤字・脱字のために語分割が著しく変わってしまうことがある。日本語が曖昧な場合の例では、「おう。」というテキストは、返事の「おう」と動詞「追う」終止形のどちらかは判断できない。前者が明快に感動詞と解析できれば「おう」を弁別することができるが、茶筌がそこまで解析できない。この



場合、本プログラムは本則に従って「おふ」（「追ふ」「負ふ」...）にたおす。「すえる」に対し、「饅えた臭い」の「すえる」よりも「腰を据える」の「すえる」のほうをより一般的と解し、システム辞書で「すゑる」に変換するような場合もある。

## 16.4 文脈要因

本システムは文脈を判断できない。日本語は省略の許容範囲が広く、文脈で一貫性・論理性を確保する傾向が強いと思う。表現そのものを単独で一意に解釈できないことがままある。文：「君にはやって見ようがない。」は、「君にはやって見よう（との気持ち）がない」とも、「君にはやって見る手立て（見様）がない」とも正則として解釈しうる。後者とした場合旧仮名遣いになると「見やう」とすべきであるが、茶筌は前者で解析するようで、「見よう」を「見よ（未然）」と「う（助動詞）」に分割する。よって本システム本則処理も「見よう」のまま残す。文脈からどちらかは判断がつくわけであるが、システムはそこまで高度にはできていない。結果を自分の目で見直すべき所以である。

## 16.5 入力表記要因

misima 旧仮名遣い変換機能はあくまでも現代仮名遣い、新字正則表記テキストの入力を想定している。仮に旧仮名の文章を入力すると、茶筌が解析できず不当な語分割を行い、その結果もとの旧仮名遣いを「改悪」してしまう可能性が高い。お手持ちの仮名漢字変換ソフトウェアが、辞書のチューニングのなされていない状態において旧仮名遣いテキストを入力すると、とんちんかんな変換結果を返すのと同じ理屈である。misima は、これはすでに旧仮名だな、などといった判断はしない。くれぐれも上で記した仕様にに基づいていることをご理解いただきたい。そもそも、日常的に旧仮名遣い・旧字体で書いているひとがこのプログラムを使うとは思えないが、仮に「旧仮名遣い・旧字体初心者」が、より一貫性をもたせたい、不正確な部分を検証・訂正したい、などの意図で半完成の旧仮名・旧字体テキストを入力するのだとすれば、目的には適わない。もとよりそんな中途半端な旧仮名・旧字体表記の習慣はやめたほうがよい。

## 17 謝辞

このプログラムのそもそもの利用目的は、筆者自身のモチーフとして、T<sub>E</sub>X で古風な文書、漢文の引用や多言語／和文混在の文書の作成を効率よく行うことにある。文科系 T<sub>E</sub>X ユーザの利便性を考慮したつもりである。文学研究論文の作成において、ロシア語をはじめ、西欧古典文献や中国・日本の古典籍等、多様な言語文化を本来の姿で引用し、巧みに計算機で文書化することが、昔からの筆者の課題であった。計算機の活用がこうした伝統的な印刷術を追い求めるのは皮肉というほかないが、ようやく個人の計算機環境においても一定のレベルで実現できるようになった気がする。

この前提として L<sup>A</sup>T<sub>E</sub>X のコミュニティで公開されている OTF パッケージや s<sub>f</sub>kanbun パッケージなどの数多くの優れた業績がある。最後になるが、日頃お世話になっているこれらのパッケージの作者の方々には改めて謝意を申し上げたい。

本プログラムは、日本語形態素解析ソフトウェア茶筌の解析結果と、『私の国語教室』における定式化とに準拠することにより、旧仮名遣い変換を試みたものである。奈良先端科学技術大学院大学情報科学研究科自然言語処理学講座の方々、福田恆存のこれら業績に敬意を表したい。

一方で、日本語表記の多様さゆえに筆者自身で茶筌辞書そのものに相当量追加せざるをえなかったり、茶筌

の解析結果を「誤解析」として採用しなかった部分もある。また『私の國語教室』は学術的に完全な定式化とはいえ、福田が本則の「例外」としてあげた語彙だけでは不十分であり、相当数の語を misima 辞書に追加しなければならなかった。このため変換精度は最終的には misima の責任であり、ユーザ自身で評価いただくしかない。

TeX 拡張変換のためのヴェトナム語、グルジア語及び機種依存文字用の変換テーブルは大友康寛氏<sup>\*31</sup>より提供いただいたものである。大友氏に感謝申し上げる。

## 18 利用条件

misima パッケージの利用条件を以下に記す。

1. 作者並びにその関係者は本パッケージの運用に関して、いかなる責任も負わない。またいかなる保障も行わない。利用者のリスクで使用するものとする。
2. 個人での利用は自由であり、無償とする。
3. 以下の形態での利用を禁止する。
  - 商用での利用
  - 企業での組織的利用
  - 人権、人命に悪影響を及ぼす危惧のある利用
  - 公序良俗に反する利用
4. 個人の Web サイトで本パッケージ配布物を一般公開する場合、作者題記電子メールアドレス宛に連絡を行うものとする。
5. misima の著作権は著者に帰属するものとする。
6. 改変は利用者の自由とする。
7. パッケージは作者が公開するものと同等の場合に限り、自由に再配布を行ってよいものとする。改変されたものを配布・公開する場合は、作者まで電子メールにて連絡を行うものとする。これは本パッケージ・サーバ機能をサイト公開する場合を含む。
8. misima は、辞書の設定に基づき今昔文字鏡番号変換を行う仕様になっている。今昔文字鏡の文字番号は株式会社大修館書店の刊行する『大漢和辞典』に基づいている。今昔文字鏡のライセンス規定により、公開パッケージ辞書はこれができない設定に修正している。今昔文字鏡番号参照機能を使用する場合は、利用者にて今昔文字鏡のライセンスを取得し、文字番号の定義を行い、ライセンス規定に則って運用すること。
9. パッケージ仕様は断りなく変更する場合がある。
10. misima SOAP クライアントで筆者サイトにアクセスする場合、常にサーバが稼働しているとは限らない。またメンテナンスのため等の事由により予告なしにサーバを停止させることがある。

パッケージ仕様は筆者の使い勝手・考え方で決めており、独善的な部分もあるかと思う。バグが潜在していることも十分に考えられる。ご意見、ご指摘、ご質問があれば筆者題記電子メール宛にお知らせいただければ幸いである。

---

<sup>\*31</sup> <http://www.klavis.info/>

## 19 更新履歴

- 2004/11/03 Ver.1.0 Web 版公開
- 2005/05/13 辞書 Ver.1.0
- 2005/07/10 Ver.2.0 T<sub>E</sub>X 変換サポート (多言語, 漢文訓点)
- 2005/08/01 SOAP Webservice-0.1 パッケージ公開
- 2006/01/01 Ver.2.1 T<sub>E</sub>X 変換タイ語, 中韓拡張
- 2006/06/10 Ver.2.2 T<sub>E</sub>X 変換教会スラヴ語, T2A 拡張
- 2006/07/22 Ver.2.3 解析機能サポート
- 2006/08/17 Ver.2.3 パッケージ (本体, Web) 公開
- 2006/08/20 Ver.2.3a SOAP リソース同梱
- 2006/09/30 Ver.2.3b 旧字 JIS 範囲内変換機能追加
- 2007/01/11 Ver.2.3c 命令 i 型問題訂正
- 2007/03/15 Ver.2.3d 「でしょ・ましよ」型旧仮名変換問題訂正
- 2007/08/19 Ver.2.3e 本則「う」変換, T<sub>E</sub>X 変換訂正
- 2007/09/23 Ver.2.3f misimaSoapClient 仕様変更, ドキュメント
- 2007/10/01 Ver.2.3g SOAP 問題訂正, SOAP Webservice-2.3g パッケージ
- 2007/10/06 Ver.2.3h システム辞書旧字変換定義, 異体字定義訂正, ドキュメント訂正
- 2007/10/07 Ver.2.3i Microsoft Word SOAP クライアント

## 付録 A システム辞書：新字-旧字変換対応表（訂 5）

misima システム辞書に登録されている新字-旧字変換対応を表に示す。旧字は UTF-8 で出力できるものを原則としているが、一部 Adobe-Japan1.5 で定義されているものも含まれている。

本表の旧字出力は、文字毎に「新字（新）-旧字（旧）-CID 番号出力（C）」を示している。CID 番号出力は、misima TeX OTF パッケージ `\CID` 旧字変換で処理したものを、pTeX で組版した結果である。CID 番号出力の欄が「-」となっているものは、旧字変換定義で CID 番号参照登録のないもので仮に CID 数値参照機能を指定した場合も（旧）で出力される。UTF-8 では定義されていない「祇」（CID 番号 7659：「祇」の旧字）なども Adobe-Japan1.5 グリフで閲覧できるはずである。

UTF-8 に含まれる旧字は網羅することを心掛けた。誤り、漏れなどあればご指摘いただけると幸いである。例えば「弁」は意味によって「辯」「辨」「辦」「瓣」など複数の旧字を使い分ける。本変換対応表では「辯」で登録されている。このような使い分けは用語・用字変換で定義することが望ましい。

Ver.2.3i 登録文字数は 1,042 文字である。

No.	新	旧	C	新	旧	C	新	旧	C	新	旧	C	新	旧	C
1	亜	亞	-	啞	啞	啞	逢	逢	逢	穉	穉	-	惡	惡	-
2	蘆	蘆	蘆	压	壓	-	扱	扱	扱	飴	飴	飴	暗	暗	暗
3	偉	偉	偉	囿	圍	圍	意	意	意	為	爲	-	緯	緯	緯
4	違	違	違	遣	遣	遣	医	醫	-	磯	磯	磯	壹	壹	-
5	溢	溢	溢	逸	逸	逸	稻	稻	-	茨	茨	茨	鯛	鯛	鯛
6	飲	飲	-	淫	淫	淫	隠	隱	-	羽	羽	羽	迂	迂	迂
7	嘘	嘘	嘘	鬱	鬱	-	厩	厩	厩	閨	閨	閨	噂	噂	噂
8	運	運	運	餌	餌	餌	營	營	-	曳	曳	-	榮	榮	-
9	衛	衛	-	銳	銳	銳	益	益	益	駅	驛	-	悅	悅	悅
10	謁	謁	謁	閱	閱	閱	円	圓	-	延	延	延	援	援	援
11	沿	沿	沿	焰	焰	焰	縁	緣	縁	艶	艶	-	遠	遠	遠
12	鉛	鉛	鉛	塩	鹽	-	於	於	於	奥	奧	-	往	往	往
13	応	應	-	横	横	横	欧	歐	-	殴	毆	-	翁	翁	翁
14	襖	襖	襖	鶯	鶯	-	鷗	鷗	鷗	黄	黃	黄	温	溫	温
15	穩	穩	-	音	音	音	化	化	化	仮	假	-	価	價	-
16	禍	禍	禍	花	花	花	貨	貨	貨	迦	迦	迦	過	過	過
17	蚊	蚊	蚊	画	畫	-	芽	芽	芽	雅	雅	雅	餓	餓	餓
18	会	會	-	回	回	-	壞	壞	-	廻	廻	廻	悔	悔	悔
19	恢	恢	恢	懷	懷	-	拐	拐	拐	晦	晦	晦	海	海	海
20	灰	灰	灰	絵	繪	-	害	害	害	慨	慨	慨	概	概	概
21	碍	礙	-	蛎	蠣	-	鈎	鉤	-	拡	擴	-	攪	攪	攪
22	殻	殼	-	覚	覺	-	較	較	較	隔	隔	隔	学	學	-
23	岳	嶽	-	楽	樂	-	割	割	割	喝	喝	喝	渴	渴	渴
24	葛	葛	葛	褐	褐	褐	轄	轄	轄	鞆	鞆	鞆	竈	竈	-

No.	新	旧	C	新	旧	C	新	旧	C	新	旧	C	新	旧	C
25	鎌	鎌	鎌	囁	囁	囁	粥	鬻	-	寒	寒	寒	勸	勸	-
26	卷	卷	-	寬	寬	寬	飲	歡	-	漢	漢	漢	灌	灌	-
27	環	環	環	緩	緩	緩	缶	罐	-	翰	翰	翰	觀	觀	-
28	諫	諫	-	還	還	還	関	關	-	陷	陷	-	韓	韓	韓
29	館	館	館	館	館	館	巖	巖	-	翫	翫	翫	顏	顏	-
30	危	危	危	器	器	器	奇	奇	-	既	既	既	期	期	期
31	帰	歸	-	氣	氣	-	祈	祈	祈	徽	徽	徽	起	起	起
32	飢	飢	飢	龜	龜	-	偽	偽	-	戲	戲	-	犧	犧	-
33	祇	祇	祇	喫	喫	喫	却	卻	-	脚	腳	-	虐	虐	虐
34	逆	逆	逆	及	及	及	吸	吸	吸	急	急	急	汲	汲	汲
35	笈	笈	笈	級	級	級	旧	舊	-	巨	巨	巨	拒	拒	拒
36	抛	據	-	挙	舉	-	虚	虚	虚	距	距	距	俠	俠	俠
37	卿	卿	卿	峽	峽	-	強	強	強	恐	恐	恐	挾	挾	-
38	教	教	教	狹	狹	-	郷	郷	郷	響	響	響	饗	饗	饗
39	堯	堯	-	曉	曉	-	僅	僅	僅	勤	勤	勤	謹	謹	謹
40	近	近	近	俱	俱	-	区	區	-	矩	矩	矩	軀	軀	軀
41	馭	驅	-	具	具	具	虞	虞	虞	喰	喰	喰	空	空	空
42	遇	遇	遇	櫛	櫛	櫛	屑	屑	屑	靴	靴	靴	勳	勳	-
43	薰	薰	薰	祁	祁	祁	啓	啓	啓	契	契	契	徑	徑	-
44	恵	恵	-	慧	慧	慧	掲	掲	掲	携	攜	携	溪	溪	-
45	経	經	-	繼	繼	-	繫	繫	繫	莖	莖	-	荆	荆	荆
46	螢	螢	-	輕	輕	-	鷄	鷄	-	芸	藝	-	迎	迎	迎
47	擊	擊	擊	隙	隙	隙	欠	缺	-	潔	潔	潔	穴	穴	穴
48	月	月	月	儉	儉	-	倦	倦	倦	健	健	健	兼	兼	兼
49	券	券	券	劍	劍	-	圈	圈	-	嫌	嫌	嫌	建	建	建
50	憲	憲	憲	拳	拳	拳	捲	捲	捲	檢	檢	-	權	權	-
51	猷	獻	-	研	研	研	県	縣	-	肩	肩	肩	謙	謙	謙
52	遣	遣	遣	險	險	-	顛	顯	-	驗	驗	-	鹼	鹼	鹼
53	嚴	嚴	-	諺	諺	諺	戸	戸	戸	雇	雇	雇	顧	顧	顧
54	呉	吳	吳	娛	娛	娛	誤	誤	誤	交	交	交	公	公	公
55	効	效	-	巷	巷	巷	広	廣	-	恒	恆	-	慌	慌	慌
56	控	控	控	更	更	更	校	校	校	構	構	構	浩	浩	浩
57	港	港	港	溝	溝	溝	硬	硬	硬	絞	絞	絞	耕	耕	耕
58	考	考	考	腔	腔	腔	荒	荒	荒	講	講	講	購	購	購
59	鉞	鑛	-	砦	礦	-	号	號	-	翹	翹	翹	告	告	告
60	国	國	國	穀	穀	穀	酷	酷	酷	鵠	鵠	鵠	黒	黒	黒
61	腰	腰	腰	甌	甌	甌	込	込	込	鎖	鎖	鎖	彩	彩	彩
62	採	採	採	歲	歲	歲	济	濟	-	采	采	采	碎	碎	-

No.	新	旧	C	新	旧	C	新	旧	C	新	旧	C	新	旧	C
63	齋	齋	-	菜	菜	菜	剂	劑	-	冴	冴	-	桫	桫	桫
64	咲	咲	咲	削	削	削	柵	柵	柵	桜	櫻	-	冊	冊	-
65	殺	殺	殺	薩	薩	薩	雜	雜	-	阜	阜	-	鯖	鯖	鯖
66	鏞	鏞	鏞	参	参	-	惨	惨	-	栈	栈	-	珊	珊	珊
67	産	産	産	蚕	蠶	-	賛	賛	-	残	殘	-	使	使	使
68	史	史	史	姉	姊	姊	姿	姿	姿	祉	祉	祉	糸	絲	-
69	視	視	視	諮	諮	諮	資	資	資	飼	飼	飼	糸	齒	-
70	児	兒	-	次	次	次	辞	辭	-	叱	叱	叱	湿	濕	-
71	実	實	-	屢	屢	屢	舍	舍	-	写	寫	-	捨	捨	捨
72	斜	斜	斜	煮	煮	煮	社	社	社	者	者	者	遮	遮	遮
73	邪	邪	邪	勺	勺	勺	杓	杓	杓	灼	灼	灼	爵	爵	爵
74	酌	酌	酌	积	釋	-	弱	弱	弱	主	主	主	寿	壽	-
75	収	收	-	周	周	周	終	終	終	繡	繡	繡	習	習	習
76	臭	臭	臭	週	週	週	酋	酋	酋	住	住	住	從	從	-
77	渋	澁	-	獸	獸	-	縦	縱	-	祝	祝	祝	肅	肅	-
78	術	術	術	述	述	述	潤	潤	潤	巡	巡	巡	遵	遵	遵
79	処	處	-	所	所	所	暑	暑	暑	曙	曙	曙	渚	渚	渚
80	緒	緒	緒	署	署	署	薯	薯	薯	諸	諸	諸	諸	諸	諸
81	叙	敘	-	勝	勝	勝	哨	哨	哨	奨	奨	-	宵	宵	宵
82	将	將	-	尚	尙	尙	床	牀	-	廠	廠	廠	松	松	松
83	梢	梢	梢	消	消	消	涉	涉	涉	焼	燒	-	硝	硝	硝
84	祥	祥	祥	称	稱	-	肖	肖	肖	蒋	蔣	蔣	訟	訟	訟
85	証	證	-	醬	醬	醬	鞘	鞘	鞘	丈	丈	丈	乘	乘	-
86	剩	剩	-	城	城	城	壤	壤	-	嬢	嬢	-	情	情	情
87	条	條	-	浄	淨	-	状	狀	狀	畳	畳	-	穰	穰	-
88	讓	讓	-	醸	釀	-	嘱	囑	-	飾	飾	飾	触	觸	-
89	食	食	食	蝕	蝕	蝕	侵	侵	侵	唇	脣	-	寝	寢	-
90	慎	慎	-	晋	晉	-	浸	浸	浸	真	眞	-	神	神	神
91	親	親	親	進	進	進	刃	刃	-	尋	尋	尋	尽	盡	-
92	訊	訊	訊	迅	迅	迅	靱	韌	靱	酢	醋	-	凶	圖	-
93	逗	逗	逗	粹	粹	-	翠	翠	翠	衰	衰	衰	遂	遂	遂
94	醉	醉	-	随	隨	-	髓	髓	-	数	數	-	枢	樞	-
95	摺	摺	摺	瀬	瀨	瀬	成	成	成	晴	晴	晴	清	清	清
96	盛	盛	盛	精	精	精	聖	聖	聖	声	聲	-	誠	誠	誠
97	請	請	請	逝	逝	逝	青	青	青	静	靜	-	齐	齊	-
98	税	稅	稅	脆	脆	脆	隻	隻	隻	籍	籍	籍	跡	蹟	-
99	撰	攝	-	窃	竊	-	節	節	節	説	說	説	雪	雪	雪
100	絶	絕	絶	蝉	蟬	蟬	専	專	-	戦	戰	-	扇	扇	扇

No.	新	旧	C	新	旧	C	新	旧	C	新	旧	C	新	旧	C
101	撰	撰	撰	栓	栓	栓	浅	浅	-	潜	潜	-	煎	煎	煎
102	煽	煽	煽	穿	穿	穿	箭	箭	箭	織	織	-	羨	羨	羨
103	舛	舛	舛	船	船	船	詮	詮	詮	賤	賤	賤	踐	踐	-
104	選	選	選	遷	遷	遷	錢	錢	-	前	前	前	全	全	全
105	禅	禪	-	噌	噌	噌	曾	曾	-	祖	祖	祖	遡	遡	遡
106	僧	僧	僧	双	雙	-	壮	壯	-	層	層	層	搜	搜	-
107	掃	掃	掃	插	插	-	搔	搔	搔	巢	巢	巢	争	爭	-
108	瘦	瘦	瘦	窓	窗	窗	総	總	-	聡	聰	-	荘	莊	-
109	装	裝	-	送	送	送	遭	遭	遭	騷	騷	-	増	增	増
110	憎	憎	憎	臟	臟	-	蔵	藏	-	贈	贈	贈	造	造	造
111	即	卽	卽	速	速	速	属	屬	-	賊	賊	賊	続	續	-
112	揃	揃	揃	尊	尊	尊	遜	遜	遜	墮	墮	-	妥	妥	妥
113	驛	驛	驛	体	體	-	対	對	-	带	帶	-	滞	滯	-
114	腿	腿	腿	退	退	退	逮	逮	逮	隊	隊	隊	黛	黛	黛
115	鯛	鯛	鯛	台	臺	-	滝	瀧	-	啄	啄	啄	択	擇	擇
116	沢	澤	-	濯	濯	濯	琢	琢	琢	蛸	蛸	蛸	達	達	達
117	脱	脫	脱	異	異	異	竖	豎	-	辿	辿	辿	柵	柵	柵
118	狸	狸	-	鱈	鱈	鱈	樽	樽	樽	单	單	-	嘆	嘆	嘆
119	担	擔	-	歎	歎	歎	炭	炭	炭	箆	箆	箆	胆	膽	-
120	誕	誕	誕	団	團	-	弾	彈	-	断	斷	-	暖	暖	暖
121	痴	癡	-	稚	穉	-	遅	遲	-	逐	逐	逐	着	著	著
122	昼	晝	-	柱	柱	柱	注	注	注	虫	蟲	-	註	註	註
123	铸	鑄	-	駐	駐	駐	瀦	瀦	瀦	猪	豬	猪	著	著	著
124	凋	凋	凋	庁	廳	-	彫	彫	彫	徴	徵	徴	懲	懲	懲
125	朝	朝	朝	潮	潮	潮	聴	聽	-	調	調	調	勅	敕	-
126	抄	抄	抄	直	直	直	朕	朕	朕	鎮	鎮	-	墜	墜	墜
127	槌	槌	槌	追	追	追	鎚	鎚	鎚	通	通	通	塚	塚	塚
128	掴	掴	掴	辻	辻	辻	坪	坪	坪	壺	壺	壺	釣	釣	釣
129	鶴	鶴	鶴	呈	呈	呈	帝	帝	帝	庭	庭	庭	廷	廷	廷
130	禎	禎	禎	程	程	程	艇	艇	艇	通	遞	遞	鄭	鄭	鄭
131	擢	擢	擢	的	的	的	適	適	適	溺	溺	溺	迭	迭	迭
132	鉄	鐵	-	填	填	填	添	添	添	転	轉	-	顛	顛	顛
133	点	點	-	伝	傳	-	兎	兔	兔	堵	堵	堵	屠	屠	屠
134	菟	菟	菟	賭	賭	賭	途	途	途	都	都	都	砺	礪	礪
135	党	黨	-	冬	冬	冬	唐	唐	唐	塘	塘	塘	禱	禱	禱
136	盗	盜	-	涛	濤	-	灯	燈	-	当	當	-	禱	禱	禱
137	糖	糖	糖	藤	藤	藤	膳	膳	膳	逃	逃	逃	透	透	透
138	騰	騰	騰	鬪	鬪	-	導	導	導	道	道	道	徳	徳	徳

No.	新	旧	C	新	旧	C	新	旧	C	新	旧	C	新	旧	C
139	澆	瀆	瀆	独	獨	-	読	讀	-	析	櫛	櫛	突	突	突
140	届	屆	-	滯	滯	滯	遁	遁	遁	吞	吞	-	那	那	那
141	内	內	內	謎	謎	謎	灘	灘	灘	栖	栖	栖	繩	繩	-
142	難	難	難	弑	貳	-	迤	邇	邇	肉	肉	肉	乳	乳	乳
143	妊	妊	-	忍	忍	忍	認	認	認	襦	襦	襦	祢	襪	襪
144	寧	寧	寧	葱	蔥	-	粘	黏	-	囊	囊	囊	恼	惱	-
145	納	納	納	脑	腦	-	霸	霸	霸	派	派	派	糜	廢	-
146	拜	拜	-	排	排	排	牌	牌	牌	輩	輩	輩	梅	梅	梅
147	壳	賣	-	這	這	這	蠅	蠅	-	秤	秤	秤	剥	剥	剥
148	博	博	博	薄	薄	薄	迫	迫	迫	縛	縛	縛	麦	麥	-
149	函	函	-	箸	箸	箸	肇	肇	肇	澆	潑	潑	発	發	-
150	醜	醜	醜	髮	髮	-	拔	拔	-	伴	伴	伴	判	判	判
151	半	半	半	叛	叛	叛	帆	帆	帆	畔	畔	畔	繁	繁	繁
152	飯	飯	飯	挽	挽	挽	晚	晚	晚	蛮	蠻	-	卑	卑	卑
153	悲	悲	悲	扉	扉	扉	斐	斐	斐	碑	碑	碑	秘	祕	-
154	緋	緋	緋	誹	誹	誹	避	避	避	樋	樋	樋	鼻	鼻	鼻
155	柎	柎	柎	稗	稗	稗	匹	匹	匹	彦	彦	彦	逼	逼	逼
156	桧	檜	-	姬	姬	姬	媛	媛	媛	謬	謬	謬	評	評	評
157	豹	豹	豹	廟	廟	廟	病	病	病	浜	濱	-	瀨	瀨	瀨
158	貧	貧	貧	賓	賓	賓	頻	頻	頻	敏	敏	敏	瓶	瓶	瓶
159	婦	婦	婦	敷	敷	敷	普	普	普	浮	浮	浮	父	父	父
160	譜	譜	譜	侮	侮	侮	服	服	服	福	福	福	覆	覆	覆
161	扌	拂	-	仏	佛	-	分	分	分	粉	粉	粉	紛	紛	紛
162	雫	雫	雫	文	文	文	丙	丙	丙	併	併	併	塀	塀	塀
163	幣	幣	幣	平	平	平	弊	弊	弊	柄	柄	柄	並	竝	-
164	蔽	蔽	蔽	警	警	警	蔑	蔑	蔑	偏	偏	偏	変	變	-
165	篇	篇	篇	編	編	編	辺	邊	-	返	返	返	遍	遍	遍
166	便	便	便	勉	勉	勉	婉	婉	婉	弁	辯	-	舗	舗	-
167	步	步	步	穂	穂	-	簿	簿	簿	包	包	包	宝	寶	-
168	崩	崩	崩	庖	庖	庖	抱	抱	抱	朋	朋	朋	泡	泡	泡
169	砲	砲	砲	縫	縫	縫	胞	胞	胞	萌	萌	萌	蓬	蓬	蓬
170	褒	褒	-	豊	豊	-	邦	邦	邦	飽	飽	飽	鵬	鵬	鵬
171	亡	亡	亡	帽	帽	帽	忘	忘	忘	忙	忙	忙	房	房	房
172	望	望	望	冒	冒	冒	頰	頰	頰	墨	墨	墨	没	沒	-
173	翻	翻	翻	凡	凡	凡	盆	盆	盆	摩	摩	摩	磨	磨	磨
174	魔	魔	魔	麻	麻	麻	每	每	每	榎	榎	-	鱒	鱒	鱒
175	侯	侯	侯	迄	迄	迄	尽	儘	-	磨	磨	磨	万	萬	-
176	満	滿	-	脈	脈	脈	明	明	明	迷	迷	迷	免	免	免



No.	新	旧	C	新	旧	C	新	旧	C	新	旧	C	新	旧	C
177	麵	麵	麵	模	摸	-	妄	妄	妄	盲	盲	盲	耗	耗	耗
178	儲	儲	儲	默	默	-	餅	餅	餅	戾	戾	戾	粗	粗	粗
179	紋	紋	紋	弥	彌	彌	約	約	約	葉	藥	-	訳	譯	-
180	躍	躍	躍	靖	靖	靖	藪	藪	藪	鎚	鎚	鎚	愉	愉	愉
181	愈	愈	愈	癒	癒	癒	諭	諭	諭	輸	輸	輸	勇	勇	勇
182	有	有	有	猶	猶	猶	猷	猷	猷	祐	祐	祐	遊	遊	遊
183	予	豫	-	余	餘	-	与	與	-	誉	譽	-	揺	搖	-
184	曜	曜	曜	様	様	-	熔	熔	熔	耀	耀	耀	要	要	要
185	謡	謡	-	遥	遙	-	養	養	養	翌	翌	翌	翼	翼	翼
186	来	來	-	莱	萊	萊	頼	頼	頼	乱	亂	-	欄	欄	欄
187	蘭	蘭	蘭	覧	覽	-	吏	吏	吏	裏	裡	-	率	率	率
188	隆	隆	隆	竜	龍	-	旅	旅	旅	虜	虜	虜	両	兩	-
189	猟	獵	-	遼	遼	遼	緑	綠	緑	燐	磷	燐	隣	隣	隣
190	鱗	鱗	鱗	麟	麟	麟	呈	壘	-	涙	淚	涙	類	類	類
191	励	勵	-	礼	禮	-	隸	隸	-	靈	靈	-	齡	齡	-
192	曆	曆	曆	歴	歷	歴	廉	廉	廉	恋	戀	-	憐	憐	憐
193	漣	漣	漣	煉	煉	煉	簾	簾	簾	練	練	練	蓮	蓮	蓮
194	連	連	連	鍊	鍊	鍊	炉	爐	-	劳	勞	-	廊	廊	廊
195	朗	朗	朗	楼	樓	-	榔	榔	榔	籠	籠	-	蛾	蠟	蠟
196	郎	郎	郎	禄	祿	-	録	錄	録	互	互	-	湾	灣	-
197	倅	倅	-	兔	兔	兔	冉	冉	冉	冤	冤	冤	凜	凜	凜
198	匕	匕	匕	啖	啖	啖	唳	唳	唳	嘲	嘲	嘲	棚	棚	棚
199	堙	堙	堙	妍	妍	妍	媾	媾	媾	屏	屏	屏	并	并	并
200	廐	廐	廐	廢	廢	廢	悦	悅	悦	扁	扁	扁	捩	捩	捩
201	搆	搆	搆	抬	擡	-	攢	攢	攢	晰	皙	-	暨	暨	暨
202	眩	曠	-	柙	櫪	櫪	枿	枿	-	柎	柎	柎	椰	椰	椰
203	櫟	櫟	-	湮	湮	湮	滬	滬	滬	燿	燿	燿	貉	貉	-
204	獮	獮	-	瑶	瑤	-	玳	瓔	-	疇	疇	-	皓	皓	皓
205	箏	箏	-	籤	籤	-	絳	濟	-	羹	羹	-	翔	翔	翔
206	翩	翩	翩	苒	苒	-	虱	蝨	-	諛	諛	諛	躄	躄	-
207	輻	輻	-	迪	迪	迪	迺	迺	迺	遘	遘	遘	扈	扈	扈
208	钁	鐸	-	鈞	鑪	-	鞞	鞞	鞞	鶩	鵝	-	鷓	鷓	-
209	赅	赅	-	熙	熙	熙	-	-	-	-	-	-	-	-	-

## 参考文献

- [1] 松本裕治, 北内啓, 山下達雄, 平野善隆, 松田寛, 高岡一馬, 浅原正幸共著, 奈良先端科学技術大学院大学情報科学研究科自然言語処理学講座, 『形態素解析システム「茶筌」 version 2.3.3 使用説明書』, <http://chasen.aist-nara.ac.jp/chasen/doc/chasen-2.3.3-j.pdf>, 2003.8
- [2] 浅原正幸, 松本裕治共著, 奈良先端科学技術大学院大学情報科学研究科自然言語処理学講座, 『ipadic version 2.7.0 ユーザーズマニュアル』, <http://chasen.naist.jp/snapshot/ipadic/ipadic/doc/ipadic-ja.pdf>, 2003.11
- [3] 福田恆存, 『私の國語教室』, 文藝春秋, 2002, (初版 新潮社, 1958)
- [4] 府川充男, 小池和男共著, 『旧字旧かな入門』, 柏書房, 2001
- [5] 久松潜一監修, 『改訂 新潮国語辞典—現代語・古語—』, 新潮社, 1980
- [6] 諸橋轍次, 渡辺末吾, 鎌田正, 米山寅太郎編, 『新漢和辞典』, 四訂版, 大修館書店, 1975
- [7] 文部省教科書局調査課国語調査室, 『くりかえし符号の使い方〔おどり字法〕(案)』, 1946.3, (北原保雄監修, 『岩波 日本語 使い方 考え方 辞典』, 岩波書店, 2003 所収)
- [8] David A. Chappell, Tyler Jewel 共著, 長瀬嘉秀監訳, テクノロジックアート訳, 『Java Web サービス』, 2002, オライリー・ジャパン
- [9] Steve Graham, Simeon Simeonov, Toufic Boubez, Doug Davis, Gien Daniels, Yuichi Nakamura, Ryo Neyama 共著, 橋本正, 西本進, 野間克司, 福原信貴監訳, 『Java による Web サービス構築』, ソフトウェアバンクパブリッシング, 2002
- [10] 錦見美貴子, 高橋直人, 戸村哲, 半田剣一, 桑理聖二, 向川信一, 吉田智子共著, 『マルチリンガル環境の実現—X Window/Wnn/Mule/WWW ブラウザでの多国語環境』, プレンティスホール出版, 1996

以上