

misima 旧仮名遣い・旧字変換支援 利用の手引き

misima SOAP Web Service Ver.2.5

安田 功, isao@yasuda.homeip.net

2008 年 7 月 8 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.

過ギニシ薔薇八名前ノミ、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	2.4	単純変換機能	10
1.1	misima の概要	3	2.5	繰返し符号変換機能	10
1.2	開発の背景 / 目的	4	2.6	仮名反転変換機能	10
1.3	従来システムの問題点	5	2.7	TEX 変換機能	11
1.4	misima の特徴	5	2.8	十六進数形式変換機能	12
1.5	misima SOAP パッケージ内容	6	2.9	マーキング機能	13
1.6	動作環境	6	2.10	無変換指定機能	13
1.7	本文書の構成	8	2.11	解析結果トレース機能	13
2	misima 機能	9	3	インストール	15
2.1	旧仮名遣い変換機能	9	3.1	Java クライアント	15
2.2	旧字変換機能	9	3.2	GNU Emacs	18
2.3	用字・用語変換機能	10	3.3	Microsoft Word	20
			3.4	秀丸エディタ	20
			3.5	Jedit X エディタ	23
			3.6	TEXShop エディタ	24

4	使い方	30	8	繰返し符号変換仕様	52
4.1	Web ブラウザ	30	9	TeX 変換仕様	52
4.2	misima オプション書式	32	9.1	多言語文字変換	52
4.3	Java クライアント	32	9.2	漢文訓点変換	68
4.4	GNU Emacs	36	9.3	くノ字点変換	70
4.5	Microsoft Word	36	10	十六進数形式変換仕様	71
4.6	秀丸エディタ	39	11	変換精度の制約・限界	73
4.7	Jedit X エディタ	39	11.1	辞書要因	73
4.8	TeXShop エディタ	40	11.2	字音仮名遣い要因	73
5	misima 処理方式	44	11.3	形態素解析要因	74
5.1	入力テキスト形式	44	11.4	文脈要因	74
5.2	日本語形態素解析	44	11.5	入力表記要因	74
5.3	変換処理シーケンス	44	12	misima SOAP プログラミング	75
6	misima 辞書	44	12.1	Java アプリケーション	75
6.1	旧字変換定義	45	12.2	Emacs Lisp	78
6.2	旧仮名変換定義	46	12.3	秀丸マクロ	78
6.3	用字・用語変換定義	48	12.4	Microsoft Office	80
6.4	単純変換定義	49	12.5	AppleScript	81
6.5	misima 辞書登録時の注意事項	50	13	謝辞	84
7	旧仮名遣い変換仕様	50	14	利用条件	84
7.1	辞書変換処理	51	15	更新履歴	85
7.2	本則変換処理	51	参考文献	86	
7.3	品詞依存処理	51			
7.4	拗音・促音変換処理	51			

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，運用の結果については無保証である．誤りがある場合はご指摘いただけると幸いである．

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

1.1 misima の概要

misima 旧仮名遣い・旧字変換支援は、現代仮名遣い・常用／当用漢字^{*1}による UTF-8 テキストを歴史的仮名遣い^{*2}・旧字体／正字体／異体字^{*3}テキストに変換するソフトウェアである。多国語文字，漢文を T_EX 形式に変換する機能も備えている。

利用者は Web ブラウザで利用できるほか，利用者 PC 上のアプリケーション：Microsoft Word 2003，秀丸エディタ，GNU Emacs (Meadow) エディタ，Jedit X エディタ，T_EXShop エディタで作成している文書中のテキストを，筆者の misima サーバで旧仮名遣い・旧字・T_EX 変換した結果テキストで置換することができる。UNIX 仮想端末，Mac OS X ターミナル，Windows コマンドプロンプトから変換させることもできる。

Mac OS X ターミナルでの misima 変換例を図 1 (3 頁) に，出力を pT_EX で組版した結果を図 2 (4 頁) に示す。これは旧仮名遣い変換，旧字変換，用字・用語変換，T_EX 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] % java misima.misimaSoapClient -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}査委員會で，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據すべきであつて\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亂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

misima システムは，misima サーバと，利用者側の PC で動作するクライアントとから構成される。misima サーバは misima の各種変換機能を実現するエンジンであり，筆者のサイト <http://yasuda.homeip.net/> に設置されている。misima SOAP Web Service (以下 misima SOAP) は，利用者がインターネット経由，SOAP インタフェースで misima サーバに接続して misima 変換機能を使うためのクライアントソフトウェア群である。

*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。

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、當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卷（昭和五十四年刊、全二十一卷）で讀んで非常に興味深いものがあつた。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據すべきであつて音聲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亂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

misima SOAP は、利用者が文書作成を行っているアプリケーションと連携し、これそのものから misima サーバに処理要求を行い、misima 変換結果を当該アプリケーションの文書に反映させる。Web ブラウザとの間でカット&ペーストでデータのやりとりを行う手間がなく、直接アプリケーション上で変換操作ができるメリットがある。

SOAP (Simple Object Access Protocol) とは 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。^{*4} SOAP Web Service 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つて、Web ブラウザ、HTML データ形式を前提としない。

本文書は misima の機能、仕様、misima SOAP クライアントソフトウェアのインストールとその利用方法について示す。misima サーバの構成要素については、利用者から見えない部分であるため、変換機能・仕様以外の説明は行わない。

misima SOAP Web Service パッケージアーカイブの最新版 `misimaSoapService-2.5.zip` は <http://yasuda.homeip.net/dl/dl.html> からダウンロードできる。

1.2 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大な引用をする必要に迫られたことにほかならない。国文学の研究論文などで、明治・大正・戦前昭和の文学の原典から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・第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。^{*5} 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、

^{*4} SOAP 仕様については W3C サイト <http://www.w3c.org/TR/SOAP/> を参照のこと。SOAP に関する詳しい解説は文献 [8, 9]などを参照のこと。

^{*5} Mac OS X, Windows XP/Vista では Unicode 文字への変換もできるようになっている。

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。

1.3 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。

- 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・第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あり、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。
- パ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。
- Web や専用 GUI のユーザインタフェースがほとんどであり、他のアプリケーションとの連携が難しい。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呵成に変換し、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。

1.4 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 1.3 節 (5 頁) で述べた問題点へ解であり、misima 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・第二水準の範囲を越えた文字を取り扱うことができる。^{*6}

Adobe-Japan1 文字空間の活用

^{*6} misima Web 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。

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。

多言語 $\text{T}_{\text{E}}\text{X}$ 対応

学術論文という成果物への指向として、筆者は $\text{T}_{\text{E}}\text{X}$ 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 $\text{pT}_{\text{E}}\text{X}$ 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 $\text{T}_{\text{E}}\text{X}$ sfkanbun 形式に変換する機能、和文繰返し表現を自動的に $\text{T}_{\text{E}}\text{X}$ 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。 $\text{T}_{\text{E}}\text{X}$ 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。またアプリケーションから呼び出して利用することも容易である。

SOAP 対応

SOAP Web Service インタフェースを実装しており、Web ブラウザだけでなく、利用者 PC 上の GNU Emacs、Microsoft Word 等のアプリケーションから遠隔にある misima サーバ機能を利用できる。選択文字列の取得/置換、外部コマンド起動のマクロインタフェースをサポートしたアプリケーションであれば、misima SOAP を呼び出して変換結果を利用できる。SOAP インタフェースにより利用者の型・目的に応じてアプリケーションの拡張が可能である。

1.5 misima SOAP パッケージ内容

misima SOAP アーカイブの内容を表 1 (7 頁) に示す。

1.6 動作環境

misima SOAP 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。

- Java クライアントの実行には、JDK 5.0(バージョン 1.5.0)以降の Java 実行環境(JVM: Java Virtual Machine)が必要である。`java -version` コマンドを投入して、`java version "1.5.0_07"` のように 1.5.0 以上のバージョン番号が返却されればよい。もしバージョンが古い場合は、サン・マイクロシステムズの Java ダウンロードサイト^{*7}から 5.0 以上の JRE (Java Runtime Environment) を

^{*7} <http://java.sun.com/javase/downloads/index.jsp>

表 1 misima SOAP アーカイブ内容

リソース	内容
README.jis	パッケージ概要説明
LICENSE	利用条件
misima-2.5.jar	misima SOAP Java クラスライブラリ
misimaSoapConnector.wsdl	misima SOAP サーバ WSDL
build/	Java クライアントビルドリソース
build/api	Java クラスライブラリ API ドキュメント
build/classes	Java クラス
build/misima	Java ソース
emacs/	GNU Emacs/Meadow 用 Emacs Lisp
hidemaru/	秀丸エディタ用マクロ
word/	Microsoft Word 2003 用文書テンプレート
shells/	Java クライアントラッパースクリプト
misima-SOAP-AppleScript.dmg	Jedit X , TeXShop 用 AppleScript

ダウンロードしてインストールする。ソースから misima SOAP Java クライアントを再構築する場合は J2SE (Java 2 Platform, Standard Edition) 5.0 以上の 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 をインストールしておく。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 環境の設置については、インターネット・リソースや JRE, SDK に関するサン・マイクロシステムズのドキュメントを参照のこと。

- UNIX 系オペレーティングシステムについては FreeBSD 6.2-RELEASE, Mac OS X Tiger 10.4.10 で Java クライアントの動作を確認している。Linux でも動作すると思われるが、未確認である。
- Emacs Lisp は、FreeBSD 6.2-RELEASE Emacs 22.0.50 及び Mac OS X 10.4.10 X11-Emacs 22.0.50, Windows XP Meadow 3.00:20050617-1 (Emacs 22.0.50 相当) で試験している。Carbon-Emacs は未確認である。Unicode フォントを利用可能にしておく必要がある。^{*8}
- 秀丸エディタ用 misima SOAP クライアントは秀丸バージョン 6.09 にて試験・動作確認を行っている。最近のバージョンであれば動作すると思う。
- Emacs, 秀丸エディタ用クライアントは Java クライアントを利用しており、上記 Java 実行環境を必要とする。
- Microsoft Word クライアントの利用には、Microsoft Word 2003 以降、Microsoft Office XP Web Service Toolkit 2.0 が必要である。Java 実行環境は不要である。
- Windows については Windows 2000, Windows XP で動作する。Windows 95/98/Me については確

^{*8} Mac OS X, Windows での Emacs Unicode 環境の調整については次に纏めた。「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>。

認していないが、動作しないと思われる。筆者は Windows XP Professional SP2 で動作確認を行っている。

- Jedit X エディタ用クライアントについては、Jedit X バージョン 1.42 にて試験・動作確認している。Jedit X の AppleScript 対応版が必要である。Java 実行環境は不要である。
- T_EXShop エディタ用クライアントについては、T_EXShop バージョン 2.14 にて試験・動作確認している。Java 実行環境は不要である。
- Jedit X 用、T_EXShop 用のクライアントは AppleScript において `call soap` 命令を使用しており、これをサポートする Mac OS X バージョンが必要である。Mac OS X 10.2 Jaguar 以降なら動作すると思われる。筆者は Mac OS X 10.4.10 Tiger で確認を行っている。
- T_EX 変換機能の変換結果を組版するために pL^AT_EX 2_ε、DVIWARE、関連スタイル/言語パッケージが必要である。misima の全 T_EX 変換機能をカバーする L^AT_EX パッケージについては表 6 (12 頁) に一覧を示している。^{*9}

1.7 本文書の構成

misima の各クライアントのインストールと使い方を調べるには、表 2 (8 頁) に示す節を参照すればよい。2 節 (9 頁) はすべてのクライアントに共通する機能を概説している。misima SOAP クライアント作成については 12 節 (75 頁) で述べている。misima 仕様詳細は 5 節 (44 頁) から 11 節 (73 頁) にかけて説明している。

表 2 クライアント説明

クライアント	該当節 (頁)
Web ブラウザ	節 4.1 (30 頁)
Windows コマンドライン	節 3.1.1 (15 頁), 節 4.2 (32 頁), 節 4.3.1 (33 頁), 節 4.3.3 (34 頁), 節 4.3.4 (35 頁)
UNIX コマンドライン	節 3.1.2 (16 頁), 節 4.2 (32 頁), 節 4.3.2 (34 頁), 節 4.3.3 (34 頁), 節 4.3.4 (35 頁)
Meadow	Windows コマンドラインと同, 節 3.2.1 (19 頁), 節 4.4 (36 頁)
Emacs	UNIX コマンドラインと同, 節 3.2.2 (19 頁), 節 4.4 (36 頁)
Microsoft Word 2003	節 3.3 (20 頁), 節 4.5 (36 頁)
秀丸エディタ	Windows コマンドラインと同, 節 3.4 (20 頁), 節 4.6 (39 頁)
Jedit X エディタ	節 3.5 (23 頁), 節 4.7 (39 頁)
T _E XShop エディタ	節 3.6 (24 頁), 節 4.8 (40 頁)

^{*9} pL^AT_EX 2_ε 及びその関連ソフトウェアの基本事項、スタイルファイルの使い方、多言語環境の解説については文献 [10] を参照。

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 5 節 (44 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 3 (9 頁) に示す．

表 3 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	“”形式で旧字を出力する．FFFF は十六進数．
T _E X 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．
T _E X OTF CID 数値参照	“\CID{99999}”形式で旧字を出力する．99999 は十進数．
T _E X 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 の公開旧字変換辞書 (6.1 節・45 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．T_EX 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 (2.9 節・13 頁) の対象外である．

Web ブラウザがクライアントの場合，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」(以上一ツ点)「々」(同ノ字点)に変換する．

マーキング (2.9 節・13 頁) の対象外である．

「々」文字を「 (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 4 (10 頁) に示す．

表 4 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	ばゝはハゝゝと笑つた
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．

「い」「い」「い」「い」はそれぞれ「づぁ」「づい」「づえ」「づぉ」に変換する．

マーキング (2.9 節・13 頁) の対象外である．

例を表 5 (11 頁) に示す .

表 5 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	過ギニシ薔薇八名前ノミ

2.7 T_EX 変換機能

多言語ならびに縦組和文・漢文を組版したい T_EX ユーザの支援を目的としている .

マーキング (2.9 節・13 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_EX 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 .^{*10}

- 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_EX 変換機能 T_EX 出力の全範囲を L^AT_EX で処理するためには , 使用する機能に応じて表 6 (12 頁) に示す各 L^AT_EX 外部パッケージが必要である . CTAN (Comprehensive T_EX 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_EX システムにインストールしておく . 「 (L^AT_EX 標準) 」としたものは teT_EX をインストールすれば標準で組み込まれるものである .^{*11}

^{*10} T_EX 変換機能の利用については , 拙稿「 misima T_EX 変換の活用 」 http://yasuda.homeip.net/misima/misima_tex.html も参照いただきたい .

^{*11} 土村氏による p_tetex3 , 角藤先生による W32T_EX でも標準で組み込まれる .

表 6 T_EX 変換前提 L^AT_EX パッケージ

OTF	和文 OpenType Virtual Fonts とスタイル ^{*1}
Babel	マルチリンガル (L ^A T _E 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 ^A T _E X 標準)
Unicode	Unicode 入力, Cyrillic T2D エンコーディングサポート ^{*2}
Teubner	ギリシア古典文献処理 ^{*3}
ThaiL ^A T _E X	タイ語 ^{*4}
VnT _E X	ベトナム語 ^{*5}
mxedruli	ゲルジア語 ^{*6}
sfkanbun	漢文訓点 ^{*7}
OldSlav	教会スラヴ語 ^{*8}
Marvosym	記号 ^{*9}
Textcomp	記号 (L ^A T _E X 標準)

^{*1} <http://psitau.at.infoseek.co.jp/otf.html>

^{*2} <http://www.tug.org/tex-archive/macros/latex/contrib/unicode/>

^{*3} <http://www.ctan.org/tex-archive/macros/latex/contrib/teubner/>

^{*4} <http://linux.thai.net/plone/TLWG/thailatex/>

^{*5} <http://vntex.sourceforge.net/download/vntex/>

^{*6} <http://www.tug.org/tex-archive/fonts/georgian/mxedruli/>

^{*7} <http://homepage3.nifty.com/xymtex/fujitas2/texlatex/>

^{*8} <http://yasuda.homeip.net/oldslav/oldslavtex.html>

^{*9} <http://www.tug.org/tex-archive/macros/latex/contrib/eurofont/marvosym/>

2.8 十六進数形式変換機能

misima 変換の最終結果テキストをすべて Unicode UCS 十六進数 `0xFFFF` 形式で出力する。文字毎にコマ区切りで出力する。行末では改行コードを `0x000a` に変換し、併せて通常の改行文字を出力する。

マーキング (2.9 節・13 頁) の対象外である。

旧字変換とともに使用した例を表 7 (12 頁) に示す。

表 7 十六進数形式変換例

変換前	変換後
森鷗外	<code>0x68ee, 0x9dd7, 0x5916, 0x000a</code> (= 森鷗外 [改行])

2.9 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もしくは TeX 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 TeX 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 “<”，“>” 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．

2.10 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，TeX 多言語文字強制変換 (9.1.2 節・56 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．

2.11 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 (14 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

インストールに先だって、筆者のサイトからパッケージアーカイブ `misimaSoapService-2.5.zip` をダウンロードし、利用 PC 上で解凍・展開しておく。説明のため、ここでは Windows, UNIX においてそれぞれ `C:¥tmp¥misimaSoapService-2.5`, `~/tmp/misimaSoapService-2.5` としてパッケージが展開されたものとする。

手順における UNIX 仮想端末 (ターミナル) の操作は tcsh シェルを前提としている。この際、一般ユーザ, スーパーユーザのプロンプトを、それぞれ `%`, `#` で示している。bash を利用している場合はそのコマンド仕様に準拠すること。

3.1 Java クライアント

本パッケージにはコマンドラインから misima SOAP サーバにアクセスするための Java クライアント (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) が添付されている。

Java クラスライブラリ `misima.jar` は JDK 5.0 でコンパイルしたものであり、実行にはバージョン 5.0 以上の JRE が必要である。

本節では Java クライアントをセットアップする方法について述べる。なお、Mac OS X は UNIX に含めて記述している。

3.1.1 Windows

ここでは「コマンドプロンプト」画面からコマンドを入力することでインストールする手順を示す。ファイルのコピー操作等はエクスプローラから実行してもよい。コマンドプロンプトは Windows XP 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。

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:¥Documents and Settings¥user>`」というプロンプトは Windows 環境によって異なる場合がある。ここで“`user`”はユーザ名である。

```
C:¥Documents and Settings¥user>mkdir c:¥misima
C:¥Documents and Settings¥user>cd ¥misima
C:¥misima>
```

2. `misima-2.5.jar` のコピー

`misima-2.5.jar` を、`C:¥misima¥misima.jar` にコピーする。

```
C:¥misima>set PKGDIR=c:¥tmp¥misimaSoapService-2.5
C:¥misima>copy %PKGDIR%¥misima-2.5.jar misima.jar
```

3. バッチファイル `misima.bat` の修正

`shells¥misima.bat` の `MISIMAJARDIR` 変数を、インストール・フォルダ名に修正する。
`C:¥misima` の場合は不要である。

4. `misima.bat` のコピー

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

```
C:¥misima>copy %PKGDIR%¥shells¥misima.bat .
```

5. 実行パスの設定

インストール・フォルダ `C:¥misima` 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。

6. クラスパスの登録

Java クラスパスに `C:¥misima¥misima.jar` を登録する。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;...`”のように“`;`”(セミコロン)で区切って jar ファイル、クラス・フォルダの絶対パスを記述する(図 4・17 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、`misima.jar` が図 5 (17 頁)のようにきちんと表示されるか確認する。

3.1.2 UNIX

1. `misima-2.5.jar` ファイルのコピー

`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-2.5.jar` を `misima.jar` のファイル名でコピーする。本説明ではそのディレクトリ名を `/home/user/lib` と仮定して説明する。説明中のディレクトリ名をユーザの事情に合わせて読み替えて指定しなければならない。

```
% cd ~
% pwd
/home/user
% setenv PKGDIR ~/tmp/misimaSoapService-2.5
```

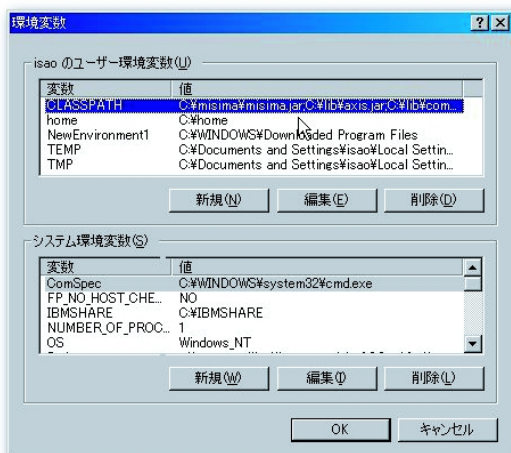



図 4 CLASSPATH 環境変数設定

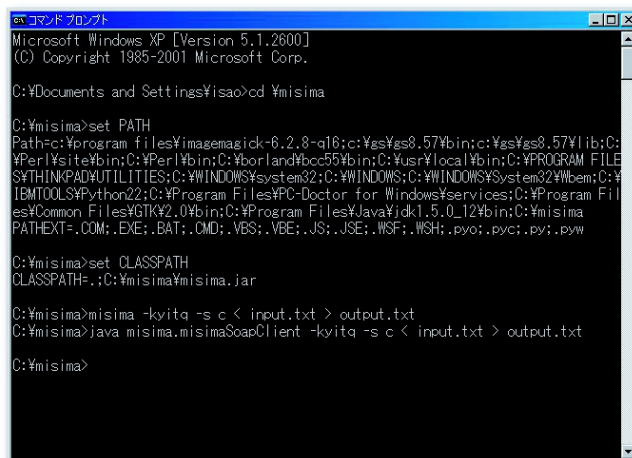


図 5 CLASSPATH 環境変数確認 / misima SOAP の実行

```
% mkdir -p lib
% cp -p $PKGDIR/misima-2.5.jar lib/misima.jar
```

2. シェル misima.sh の修正

shells/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。

```
MISIMAJARDIR=/home/user/lib
```

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。

```
% cp -p $PKGDIR/shells/misima.sh ~/bin
```

4. CLASSPATH 環境変数の設定

misima.jar 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。

3.1.3 Java コンパイル環境

Java クライアントは、すでにコンパイル済みのクラスモジュールが misima.jar に含まれている。

再コンパイルを行う場合は、表 8 (18 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。パッケージには Ant ビルド用の build.xml を添付している。

表 8 必要クラスライブラリ

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>java-getopt-1.0.13.jar</code>	GNU getopt Java port ^{*1}

^{*1} <http://www.urbanophile.com/arenn/hacking/download.html>

3.1.4 Java 環境

misima SOAP Java クラスライブラリは表 9 (18 頁) に示すオープンソース・クラスライブラリを利用している。`.misima-2.5.jar` に同梱しているので、misima SOAP 機能を利用する限りにおいて別途ダウンロード/設定を行う必要はない。

表 9 `misima.jar` 同梱クラスライブラリ

ライブラリ	ライセンス	備考
<code>axis.jar</code>	Apache 2.0 ^{*1}	Axis 1.4 添付
<code>jaxrpc.jar</code>	JDL 1.0 ^{*2}	Axis 1.4 添付
<code>saaj.jar</code>	CDDL 1.0 ^{*3} & GPL 2 ^{*4}	Axis 1.4 添付
<code>log4j-1.2.8.jar</code>	Apache 2.0	Axis 1.4 添付
<code>wsdl4j-1.5.1.jar</code>	CPL 0.5 ^{*5}	Axis 1.4 添付
<code>commons-logging-1.0.4.jar</code>	Apache 2.0	Axis 1.4 添付
<code>commons-discovery-0.2.jar</code>	Apache 2.0	Axis 1.4 添付
<code>java-getopt-1.0.13.jar</code>	Lib GPL 2 ^{*6}	GNU getopt Java port ^{*7}

^{*1} Apache License Version 2.0

^{*2} Java Distribution License Version 1.0

^{*3} Common Development and Distribution License Version 1.0

^{*4} GNU General Public License Version 2

^{*5} Common Public License Version 0.5

^{*6} GNU Library General Public License Version 2

^{*7} <http://www.urbanophile.com/arenn/hacking/download.html>

3.2 GNU Emacs

GNU Emacs は UNIX ワールドにおいて vi とともにテキストエディタの双壁である。Emacs Lisp (以下 Elisp) によるプログラミング機能・拡張性と多国語への対応とにおいて類をみないアプリケーションである。

Windows においても Meadow という名称で移植されている。

ここでは Windows Meadow 3.00 , UNIX Emacs 22.0.50 向け misima クライアントのインストール・設定の手順を示す。

Emacs 用クライアントは内部で Java クライアントを利用しており , Java クライアントを予め導入しておく (3.1 節・15 頁参照) 。

3.2.1 Windows

1. Elisp のインストール

`emacs¥misimasoap.el` を Elisp 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 (ここでは `¥meadow` とする) にある `site-lisp` フォルダに格納すればよい。

```
C:¥misima>copy %PKGDIR%¥emacs¥misimasoap.el ¥meadow¥site-lisp
```

2. `.emacs` の設定

Meadow 初期設定ファイル `.emacs` に以下の 2 行を追加する。`misimasoap-path` は `misima.bat` 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")
```

3.2.2 UNIX

1. Elisp のインストール

`misimasoap.el` をバイトコンパイルし , Elisp 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 $PKGDIR/emacs
% emacs -batch -f batch-byte-compile misimasoap.el
% sudo cp -p misimasoap.elc /usr/local/share/emacs/site-lisp
```

2. `.emacs` の設定

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")
```

3.3 Microsoft Word

本節では Microsoft Word 2003 用クライアントのセットアップ方法を説明する。Microsoft Office XP Web Services Toolkit 2.0 のインストールと、本パッケージ文書テンプレートの所定位置への格納とを実施すればよい。

3.3.1 Microsoft Office XP Web Services Toolkit 2.0

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。

1. Microsoft Office Web サイト <http://www.microsoft.com/japan/office/previous/xp/developer/webservices/download.msp>(図 6・21 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 7・21 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 8・21 頁)して、Web Service References Tool が表示されればよい(図 9・22 頁)。

3.3.2 Word 文書テンプレートの格納

misima Word クライアント VBA (Visual Basic for Applications) コード及び misima SOAP 用マクロ起動メニューを備えた Word 文書テンプレート `misimaTemplate.dot` をインストールする。Office マクロウィ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制限されている場合があるので、運用を確認して設定を行うこと。

`word¥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。

3.4 秀丸エディタ

秀丸エディタは Windows ユーザの間で定評のあるエディタのひとつである。

misima SOAP 秀丸クライアントは、秀丸マクロで実現しており、マクロ内部で Java クラスライブラリと本パッケージ提供 Windows プログラムランチャをコールしている。よって秀丸独自の設定に先立って Java クライアントをインストールしておく必要がある(3.1 節・15 頁参照)。

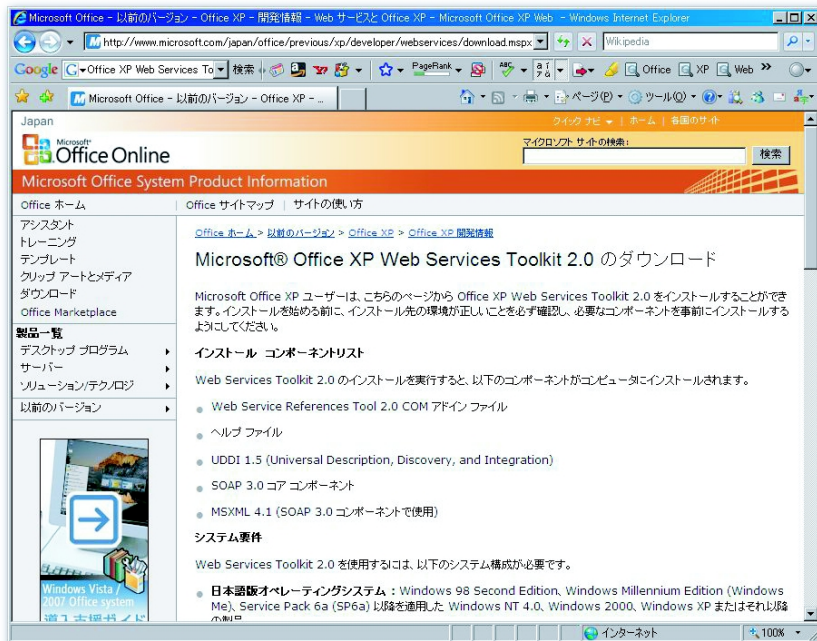


図 6 Web Services Toolkit ダウンロードサイト

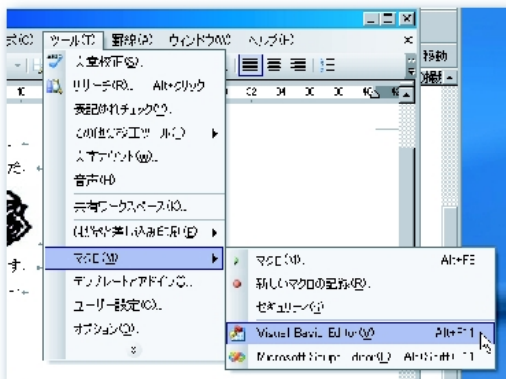


図 7 Visual Basic Editor

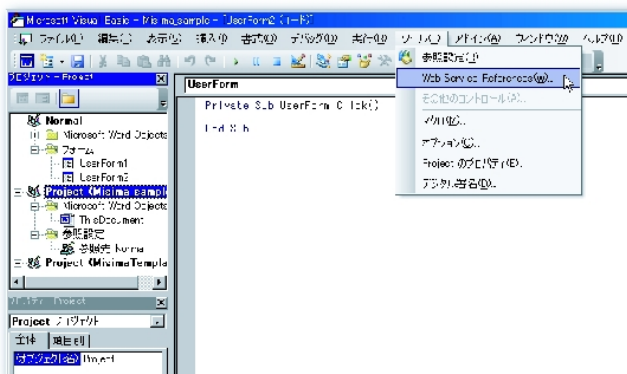


図 8 Web Service References Tool の表示

3.4.1 秀丸用マクロファイルのコピー

hidemaru フォルダ内の秀丸マクロ *.mac 一式を C:¥misima にコピーする。

```
C:¥misima>copy %PKGDIR%¥hidemaru¥*.mac .
```

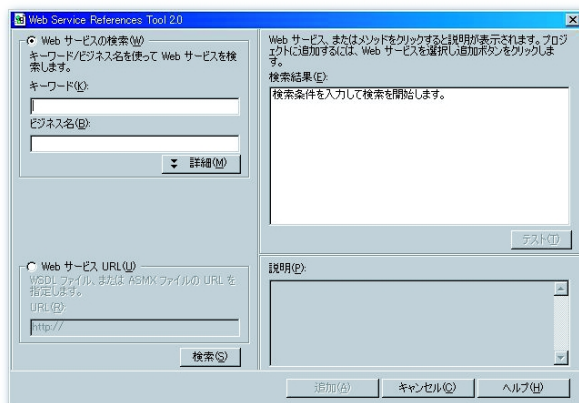


図 9 Web Services References Tool

3.4.2 misimalaunch.exe のコピー

hidemaru フォルダ内の misimalaunch.exe ファイルを C:¥misima にコピーする。

```
C:¥misima>copy %PKGDIR%¥hidemaru¥misimalaunch.exe .
```

3.4.3 マクロの登録

秀丸を起動し、「マクロ」-「マクロの登録」からマクロ登録画面（図 10・23 頁）を表示させる。「ファイル名」ボックス右にある「...」をクリックして C:¥misima 下にコピーした秀丸マクロファイルを指定する。登録の際、マクロ名（「タイトル」ボックス）やショートカットは適宜利用者が分かりやすいよう設定する。

提供する各マクロは用途に応じて misima 変換オプションが異なる。マクロファイル名と変換操作内容、misima オプションとの対応は表 10（22 頁）に示すとおりである。misima オプションについては表 13（42 頁）を参照。

表 10 秀丸マクロ一覧

マクロファイル名	変換操作内容	misima オプション
misima-kyits-c.mac	旧字・旧仮名変換	-kyit -s c
misima-kyits-tex.mac	旧字・旧仮名・T _E X 変換	-kyit -s a -x kuitx
misima-tex.mac	T _E X 変換	-x kutx
misima-ts-c.mac	旧字変換	-t -s c

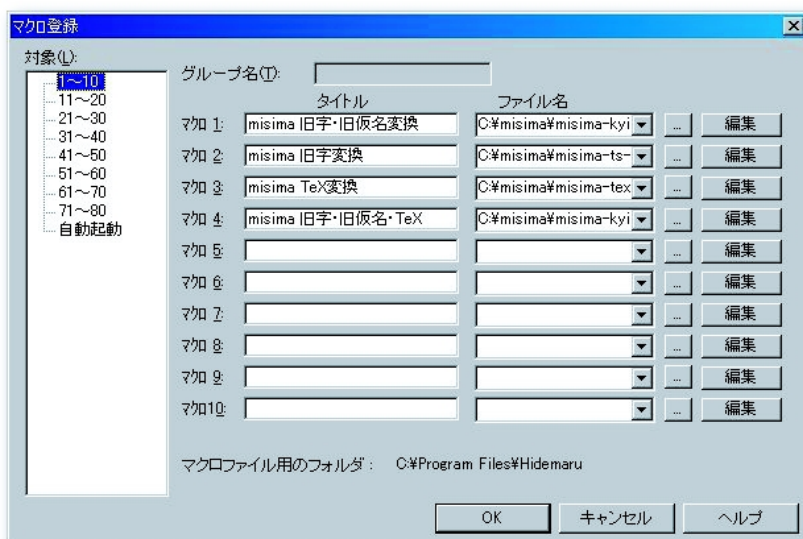


図 10 秀丸マクロ登録画面

3.5 Jedit X エディタ

Jedit X エディタは Mac OS X で動作するシェアウェアである。Jedit X 用 misima SOAP クライアントは AppleScript で実装しており、`JeditX/*.scpt` ファイル一式を Jedit X のスクリプトとして登録すれば利用可能となる。Java クライアントのインストールは不要である。

3.5.1 Jedit X AppleScript 登録

以下、順を追って導入手順を示す。

1. パッケージ添付の `misima-SOAP-AppleScript.dmg` ディスクイメージを Finder からダブルクリックしてマウントする。“Jedit X”と“TeXShop”の二つのフォルダが表示される。(図 11・24 頁)
2. ディスクイメージの“Jedit X”フォルダを開く。
3. Jedit X を起動し、メニューバーの右端にあるリボン型アイコン(スクリプトアイコン)から「スクリプトウィンドウを表示」をクリックし、スクリプトウィンドウを表示させる。(図 12・25 頁)
4. Finder “Jedit X”フォルダ内のスクリプトすべてを選択して、Jedit X スクリプトウィンドウにドラッグする。これで Jedit X 用 misima AppleScript のインストールは完了である。(図 13・26 頁)

3.5.2 AppleScript 内容

提供する各 AppleScript は用途に応じて misima 変換オプションが異なる。AppleScript ファイル名と変換操作内容、misima オプションとの対応は表 11 (24 頁) に示すとおりである。misima オプションについては表 13 (42 頁) を参照。

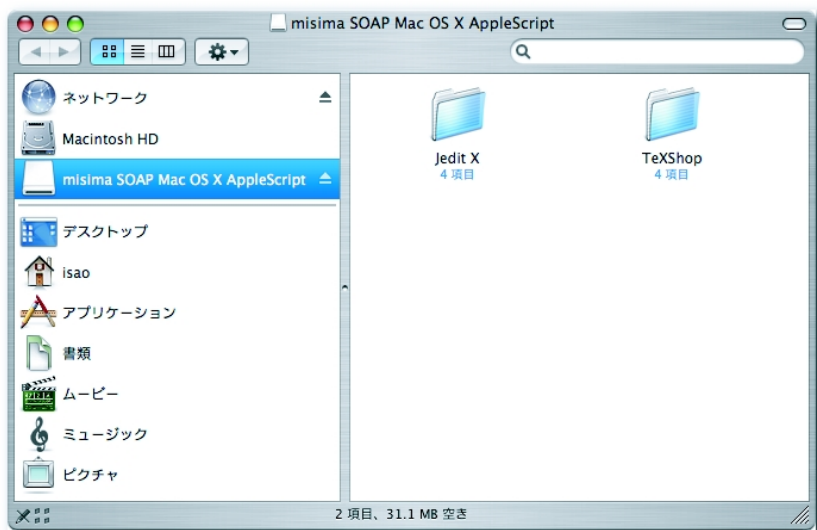


図 11 ディスクイメージのマウント

表 11 Jedit X AppleScript 一覧

AppleScript	変換操作内容	misima オプション
<code>misima 旧字旧仮名変換.scpt</code>	旧字・旧仮名変換	<code>-kithy -s c</code>
<code>misima 旧字旧仮名 TeX 変換.scpt</code>	旧字・旧仮名・TeX 変換	<code>-kithy -s a -x kuitx</code>
<code>misimaTeX 変換.scpt</code>	TeX 変換	<code>-h -x kutx</code>
<code>misima 旧字変換.scpt</code>	旧字変換	<code>-th -s c</code>

3.6 TeXShop エディタ

TeXShop は Mac OS X で動作する TeX 統合編集環境である。TeXShop 用 misima SOAP クライアントは AppleScript で実装しており、`TeXShop/*.txt` ファイル一式を TeXShop のスクリプトとして登録すれば利用可能となる。Java クライアントのインストールは不要である。

3.6.1 TeXShop AppleScript 登録

以下、順を追って導入手順を示す。

1. パッケージ添付の `misima-SOAP-AppleScript.dmg` ディスクイメージを Finder からダブルクリックしてマウントする。“Jedit X”と“TeXShop”の二つのフォルダが表示される。(図 11・24 頁)
2. ディスクイメージの“TeXShop”フォルダを開く。
3. TeXShop を起動し、メニューバーの「マクロ」-「マクロエディタを開く...」をクリックし、マクロエディタを表示させる。(図 14・27 頁)
4. 「新規マクロ」をクリックし、「項目名 :」のエリアにスクリプト名(ここでは「`misima 旧字旧仮名`」

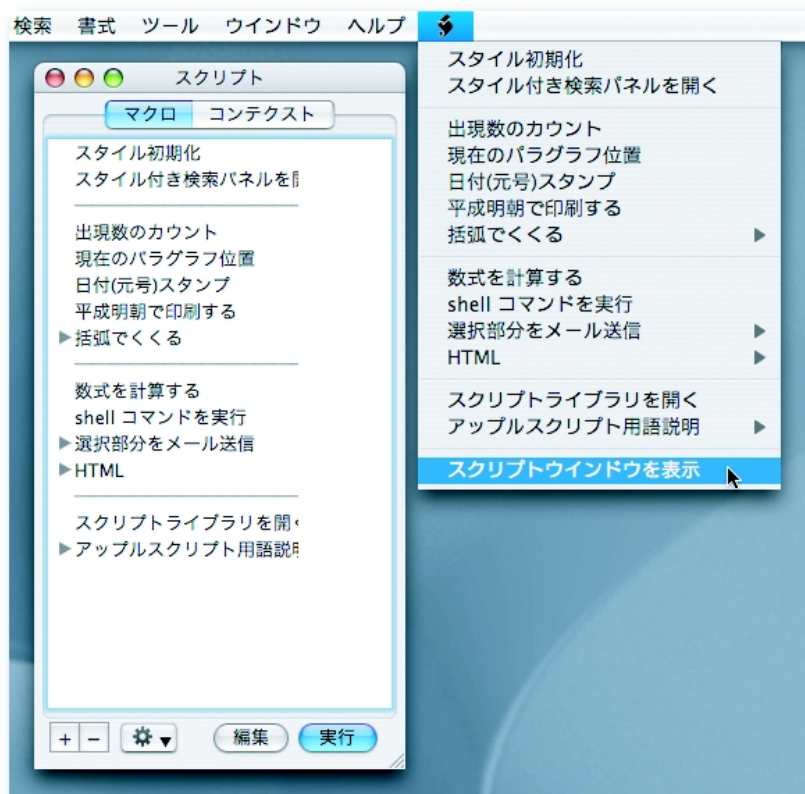


図 12 Jedit X スクリプトウインドウ

TeX 変換」) を入力する。(図 15・27 頁)

5. 「内容 :」のエリアをクリックして、カーソルを位置づける。ここで適当なエディタ (例は Jedit X) で、TeXShop フォルダにある misima AppleScript ファイル (拡張子 .txt) から「項目名 :」指定に相当するスクリプト (ここでは「misima 旧字旧仮名 TeX 変換.txt」) を開き、全内容をコピー (Command+A Command+C) する。(図 16・28 頁)
6. コピーしたスクリプト内容を、マクロエディタの「内容 :」のエリアにペースト (Command+V) する。(図 17・28 頁)
7. 「保存」をクリックすると、TeXShop にスクリプトが登録される。このときショートカットキーを適宜設定してもよい。上記操作を misima 提供スクリプト分繰り返す。
8. すべてのスクリプトを登録したら、マクロエディタ「Applescript」メニューの適当な位置に misima スクリプトを再配置しておく。項目名をマウスでドラッグして移動すればよい。これで TeXShop 用 AppleScript のインストールは完了である。(図 18・29 頁)

3.6.2 AppleScript 内容

提供する各 AppleScript は用途に応じて misima 変換オプションが異なる。AppleScript ファイル名と変換操作内容、misima オプションとの対応は表 12 (26 頁) に示すとおりである。misima オプションについては表 13 (42 頁) を参照。

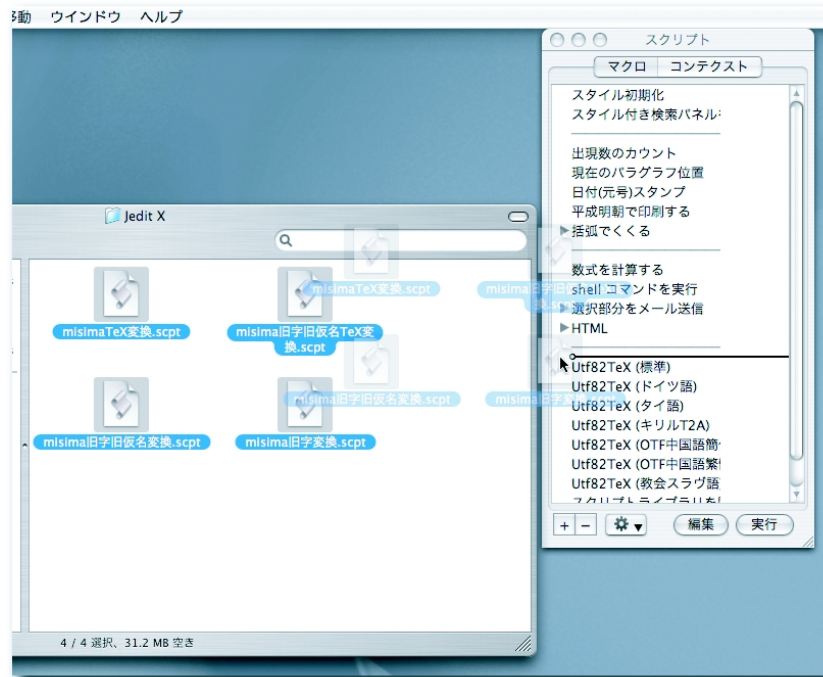


図 13 Jedit X スクリプトウィンドウへ登録

表 12 $\text{T}_{\text{E}}\text{X}$ Shop AppleScript 一覧

AppleScript	変換操作内容	misima オプション
<code>misima 旧字旧仮名変換.txt</code>	旧字・旧仮名変換	<code>-kyith -s c</code>
<code>misima 旧字旧仮名 TeX 変換.txt</code>	旧字・旧仮名・ $\text{T}_{\text{E}}\text{X}$ 変換	<code>-kyith -s a -x kuitx</code>
<code>misimaTeX 変換.txt</code>	$\text{T}_{\text{E}}\text{X}$ 変換	<code>-h -x kutx</code>
<code>misima 旧字変換.txt</code>	旧字変換	<code>-th -s c</code>

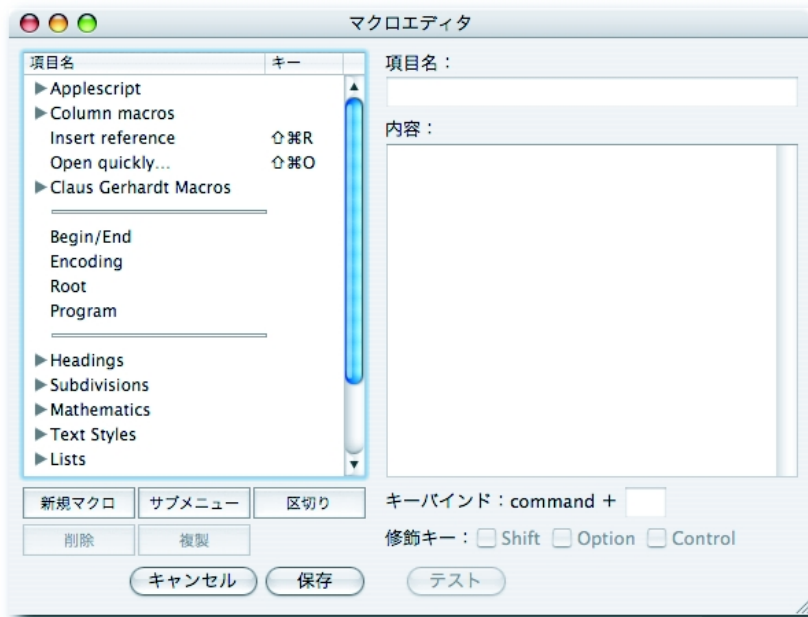


図 14 TeXShop マクロエディタ

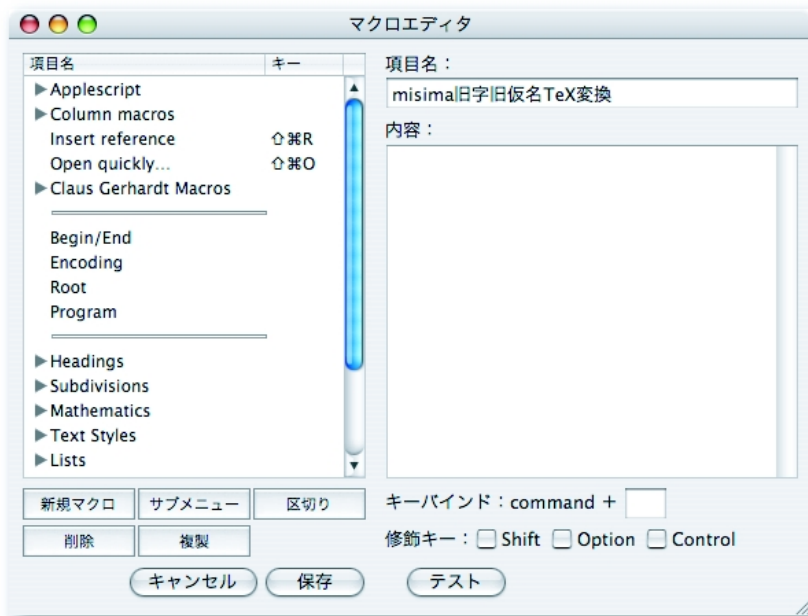
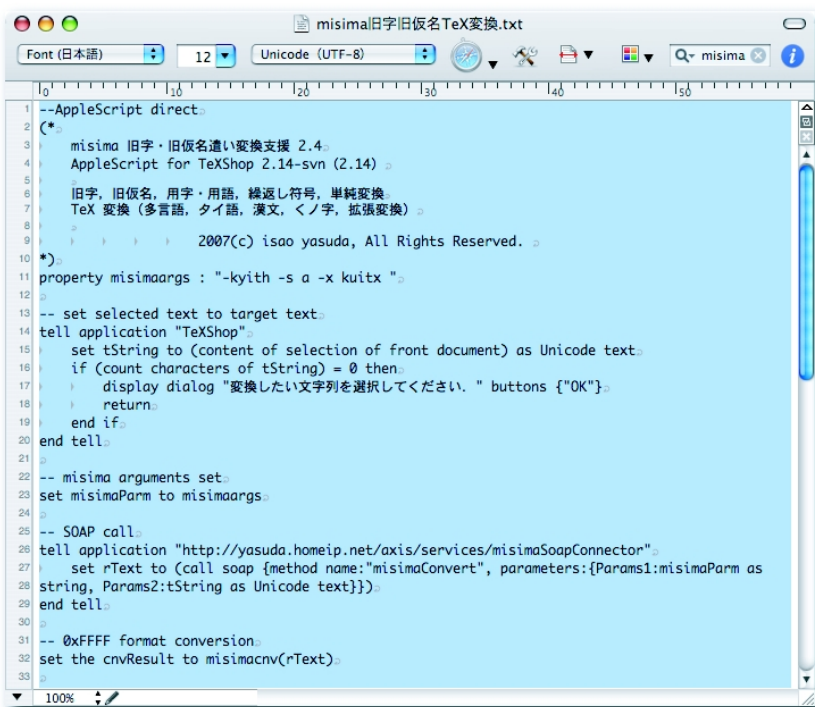


図 15 TeXShop: 項目名の入力



```

1  --AppleScript direct.
2  (*
3   misima 旧字・旧仮名遣い変換支援 2.4.
4   AppleScript for TeXShop 2.14-svn (2.14) .
5
6   旧字、旧仮名、用字・用語、繰返し符号、単純変換。
7   TeX 変換 (多言語、タイ語、漢文、くノ字、拡張変換) .
8
9   )
10 )
11 property misimaargs : "-kyith -s a -x kuitx "
12
13 -- set selected text to target text.
14 tell application "TeXShop"
15     set tString to (content of selection of front document) as Unicode text.
16     if (count characters of tString) = 0 then
17         display dialog "変換したい文字列を選択してください。" buttons {"OK"}
18     return
19     end if.
20 end tell.
21
22 -- misima arguments set.
23 set misimaParm to misimaargs.
24
25 -- SOAP call.
26 tell application "http://yasuda.homeip.net/axis/services/misimaSoapConnector"
27     set rText to (call soap {method name:"misimaConvert", parameters:{Params1:misimaParm as string, Params2:tString as Unicode text}}).
28 end tell.
29
30 -- 0xFFFF format conversion.
31 set the cnvResult to misimacnv(rText).
32
33

```

図 16 TeXShop: スクリプト内容のコピー

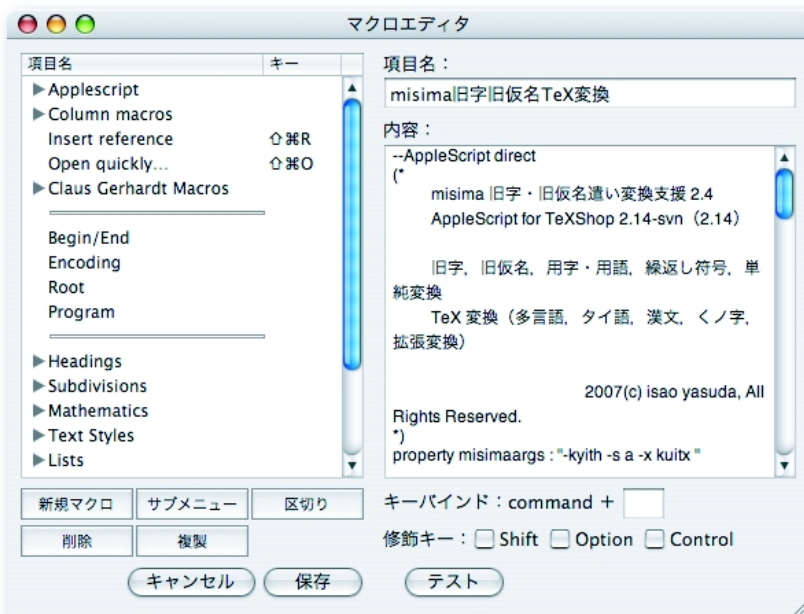


図 17 TeXShop: スクリプトの貼付け

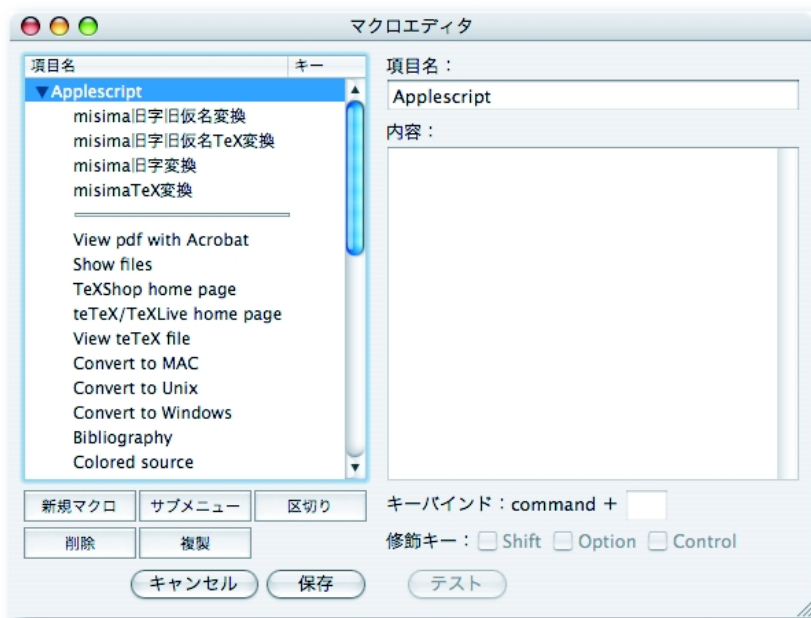


図 18 TeXShop: 追加された全スクリプト

4 使い方

ここでは Web ブラウザ, misima SOAP クライアントを用いて misima サーバ変換を行う方法を示す。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、ネットワークやサーバの状態に依存して、多少時間がかかるかも知れない。

4.1 Web ブラウザ

ここでは Web ブラウザを用いて misima 変換を実行する操作を説明する。Web ブラウザからは CGI 版と Servlet 版の二つが利用できる。CGI 版は misima の変換オプションをより細かく設定でき、ユーザ辞書の利用が可能である。一方、Servlet 版は簡易にオプションを指定し同じページに変換結果を挿入するユーザインタフェースをもつ。

CGI 版は古典的な CGI (Common Gateway Interface) プログラムとして実装されており、misima プログラム起動、オプション解析、辞書読み込み等のすべての処理をサーバで行うことと、プロセス生成、変換結果ページ生成等のオーバーヘッド要因とにより、Servlet 版に比べ著しく遅いという欠点がある。

これに対し、Servlet 版は、ユーザ辞書が使えない欠点があるが、misima オプション解析を Web ブラウザ側 JavaScript で実行することにより負荷分散を図り、Web クエリを受け付けるサーバ (Java Servlet) がすでに常駐して待ち構えている上に、misima 変換エンジン (misimaserver) が起動状態でメモリに常駐し辞書を読み込んだ直後から処理を開始でき、かつブラウザへのデータ送信が変換結果テキストのみで軽いため、CGI 版に比べて高速に動作する。

CGI 版 / Servlet 版のいずれも旧仮名遣い変換、旧字変換、用字・用語変換、T_EX 多言語標準変換、仮名反転変換を処理できる。単純変換はどちらも既定値で指定されていて、これを外すことはできない。筆者はユーザ辞書を指定する特殊な利用をしない限りは Servlet 版をお勧めする。

4.1.1 CGI 版

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 20 (32 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。

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。プロダクト管理システムが 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. T_EX 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 6 節 (??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内容を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、保存した内容を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 19・31 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 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。

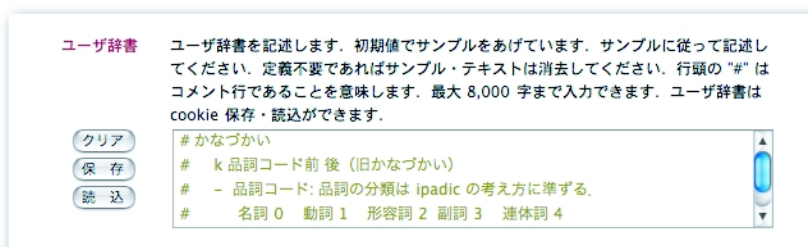


図 19 ユーザ辞書

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 21 (32 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。

4.1.2 Servlet 版

1. フォームへのアクセス

ブラウザで <http://yasuda.homeip.net/misimaservlet/misima.html> を開く。(図 22・33 頁)

2. テキストの入力・オプションの指定

プルダウンから「旧字/旧仮名/用語/繰返」等の変換オプションを選択する。旧字変換は UTF-8 出力固定である。必要に応じて「LaTeX」、「第一・第二水準内」、「仮名反転」のチェックボックスを指定する。オプション指定の意味は、節 2 を参照すれば、とくに説明を要しないと思われる。変換対象としたテキストを「テキスト入力」のエリアに入力する。オプション、対象テキストを入力したら「変換」ボタンをクリックする。処理中は「misima 処理中...」との緑色のメッセージを出力し、完了したら「変換結果」の右側に変換結果テキストが挿入される。図 23 (34 頁) に変換処理後の様子を示す。

3. オプションの保存機能

よく使用するオプションをブラウザに記憶させておくことができる。オプション選択を行った状態で「保存」ボタンを押し下げるとブラウザのクッキーに保存する。「読込」ボタンにより保存したオプションを呼び出す。「削除」は保存した内容を破棄する。

4. 旧字変換オプションの注意事項

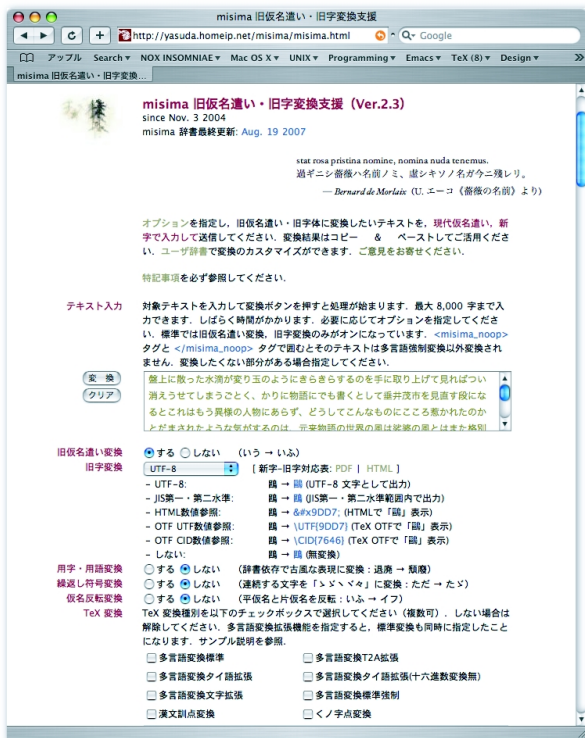


図 20 フォーム入力 (変換前)



図 21 変換結果出力

CGI 版の場合と考え方は同じである (節 4.1.1・30 頁参照)。変換結果を他のアプリケーションに貼付けたときに文字化けするようなら「第一・第二水準内」をオンにして実行する。

4.2 misima オプション書式

misima SOAP クライアント共通で指定できる misima オプション書式を図 24 (34 頁), オプション引数の意味を表 13 (42 頁) に示す。Java クライアントではコマンド引数として指定する。Microsoft Word マクロ, AppleScript ではスクリプト内の misima オプション用の変数にセットする。

4.3 Java クライアント

Java クライアントはコマンドラインから実行する。

変換対象テキストの文字数は 1 回あたり 8,000 バイトまで可能である。

ユーザ辞書機能はサポートしていない。

コマンドライン実行は二種類ある。ラッパーツール (UNIX: `misima.sh`, Windows: `misima.bat`) を利用する方法と, `java` クラスランチャでクラスモジュール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。



図 22 Servlet 版画面

4.3.1 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` 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。

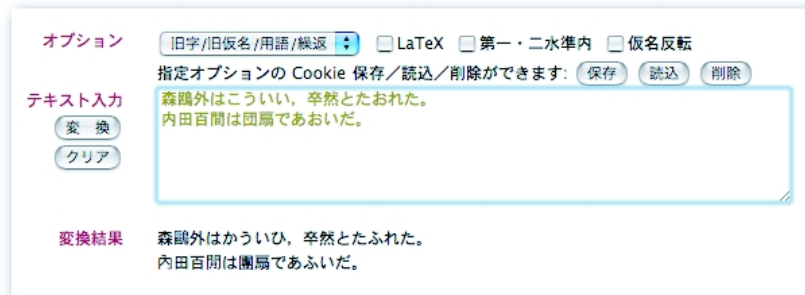


図 23 Servlet 版変換処理結果

```
[ -k|-y|-t|-i|-n|-h|-q|-d|-g|-b|-s {cjhuam}|-x <k|u|f|x|r|t|T|h|i|a>|
-c {KCT}|-m {ht}|-u URL ]
- [ ] 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 24 misima オプション書式

- `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 5 (17 頁) のはじめの実行例のとおりである .

4.3.2 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
```

4.3.3 Java ランチャ起動

上記のラッパースクリプトを用いるのではなく , `java` クラスランチャでクラスモジュールを直接コールする方法を示す .

Java クライアントには実行可能なクラスモジュールが二つある .

`misimaSoapClient.class`

標準入力から対象テキストを入力し , `misima SOAP` 変換を行い , 変換結果を標準出力に出力する .

ラッパースクリプトはこれを利用している。

`misimaSoapCbClient.class`

クリップボードのテキストを入力し、`misima SOAP` 変換を行い、変換結果をクリップボードに出力する。秀丸エディタ用 `misima SOAP` マクロのためのクラスモジュールであるが、クリップボード変換としてコマンドラインから実行してもよい。アプリケーションにおいて変換したいテキストをクリップボードにコピーしたあと、本クラスモジュールを実行する。終了したら、ペーストすると `misima` 変換結果が当該アプリケーションに貼付くはずである。

`java` コマンドのクラス指定においては、クラスモジュールの前にパッケージ名 `misima` を付加しなければならない。

UNIX での実行方法を以下に示す。

```
% java misima.misimaSoapClient -kyitq -s c \  
  < inputfile > outputfile
```

Windows においても実行方法は UNIX と同様である。Windows での Java クラスランチャによる実行の様子は、図 5 (17 頁) の二つ目の実行例のとおりである。

4.3.4 プロキシ環境

企業ネットワークのように、Squid などのプロキシサーバを経由して外部インターネットワークと接続している場合、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ことがある。

```
% 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" \  
  misima.misimaSoapClient -kyitq -s c \  
  < 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。`PROXY` 変数雛形がコメントとして予め指定されているので、利用者環境に合わせプロキシホスト名などのパラメータを変更し、コメント記号を削除して変数を有効にすればよい。`misima SOAP Emacs` クライアントをプロキシ経由で利用する際は、この変更は必須となる。

通常、個人のインターネット接続環境はプロキシを経っていないので、このプロキシ設定を行ってはない。

4.3.5 Java クライアント実行例

UTF-8 で作成したテキストファイルを Java クライアント (Java ランチャ起動, Mac OS X ターミナル) によって misima SOAP サーバで変換する実行例を示す。図 25 (36 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 misima SOAP 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。

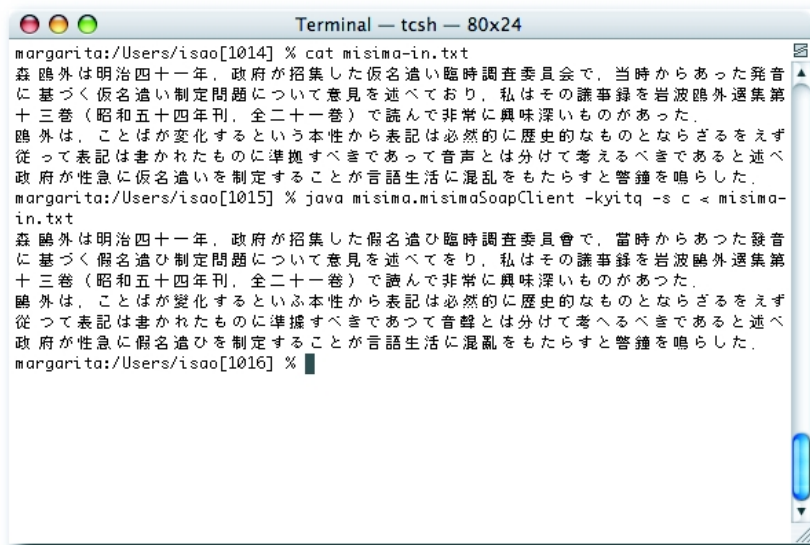


図 25 Java クライアント (変換前後)

4.4 GNU Emacs

Emacs からの misima SOAP 変換実行は “M-x misimasoap-コマンド RET” とする。UNIX Emacs , Windows Meadow で操作は同じである。

利用可能な Emacs コマンドを表 14 (43 頁) に示す。misima オプションについては表 13 (42 頁) を参照。-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。

Windows XP Meadow 3.00 環境上での変換操作を図に示す。図 26 (37 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 27 (37 頁) は変換実行後を示している。

もし変換中になんらかの障害が発生して、選択テキストが失われてしまった場合は、C-/ でアンドウーして復元させる。

4.5 Microsoft Word

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。

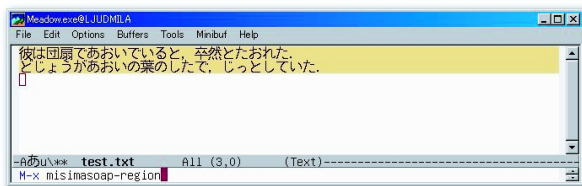


図 26 Windows Meadow 変換前

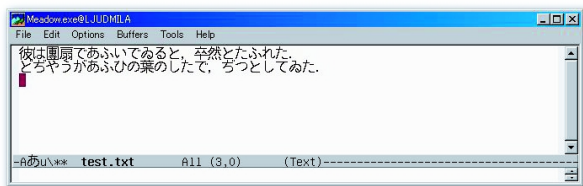


図 27 Windows Meadow 変換後

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始まる。その様子を図 28 (37 頁) に示す。

変換が終了すると選択領域が変換結果で置換される (図 29・37 頁)。

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 13・42 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 30・38 頁)。

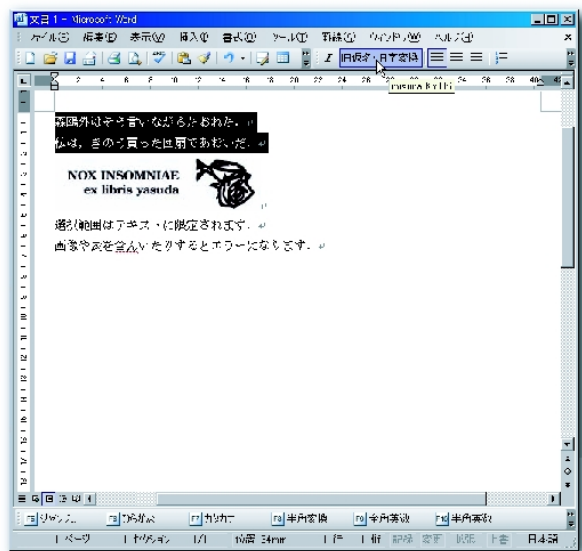


図 28 Word クライアント変換前

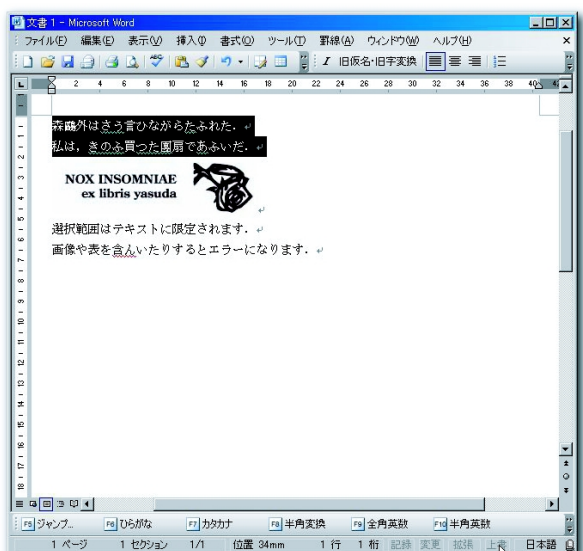


図 29 Word クライアント変換後

変換対象テキストの文字数は 1 回あたり 8,000 バイトまで可能である。

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。

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 31 (38 頁) に示すように画像・表な

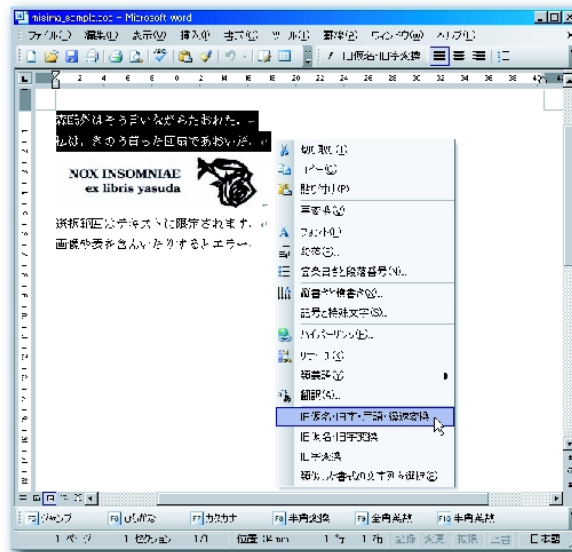


図 30 右クリックショートカットメニュー

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 32・38 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。

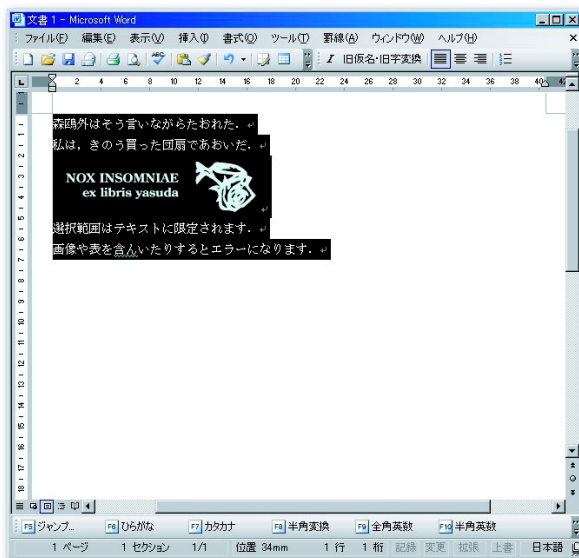


図 31 テキスト以外の選択は不可



図 32 XML パーサエラー

もし変換中になんらかの障害が発生して、選択テキストが失われてしまった場合は、「**Ctrl+Z**」で操作を取り消して復旧させる。

4.6 秀丸エディタ

秀丸エディタで文書を作成するなかで、misima 旧字・旧仮名遣い変換が必要となるテキストを選択（マウスで範囲をドラッグ）し、適当な変換オプションにバインドした misima 用マクロを実行すればよい。

マクロを実行すると、選択テキストが消えて削除されたように見えるが、しばらくすると変換結果が挿入され、結果的に選択テキストが変換結果で置換される。

テキストが選択されていないと、エラーメッセージを出力する。

変換対象テキストの文字数は 1 回あたり 8,000 バイトまで可能である。

もし、結果が変わるはずなのに変換されない場合（時折発生する）、再度変換操作を行ってみるとよい。misima 変換プロセスエラーを検知したときは元の変換対象テキストが復元されるようにしているが、万一、変換中になんらかの障害が発生してテキストが失われるような事態が起きたら、「**Ctrl+Z**」でアンドゥーして戻してほしい。

変換前後をそれぞれ図 33（39 頁）、図 34（39 頁）に示す。

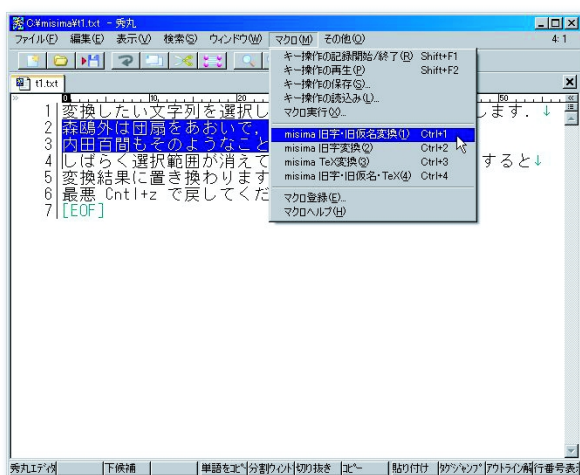


図 33 秀丸エディタ変換前

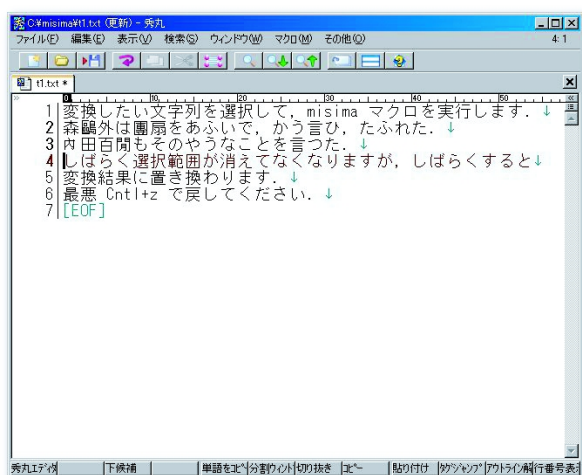


図 34 秀丸エディタ変換後

4.7 Jedit X エディタ

Jedit X エディタで文書を作成するなかで、misima 旧字・旧仮名遣い変換が必要となるテキストを選択（マウスで範囲をドラッグ）し、適当な変換オプションにバインドした misima 用 AppleScript を実行すればよい。

スクリプトを実行すると、選択テキストが変換結果で置換される。

テキストが選択されていないと、エラーメッセージを出力する。

変換対象テキストの文字数は 1 回あたり 8,000 バイトまで可能である。

もし、結果が変わるはずなのに変換されない場合（時折発生する）、再度変換操作を行ってみるとよい。万一、変換中になんらかの障害が発生してテキストが失われるような事態が起きたら、「**Command+Z**」でアンドゥーして戻してほしい。

変換前後をそれぞれ図 35（40 頁）、図 36（40 頁）に示す。

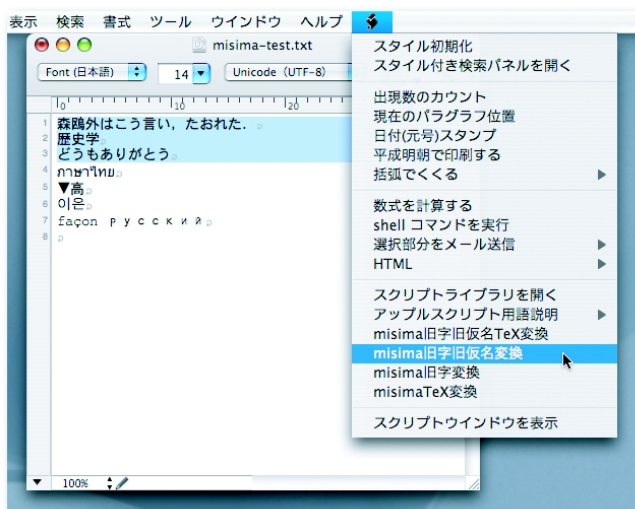


図 35 Jedit X 変換前

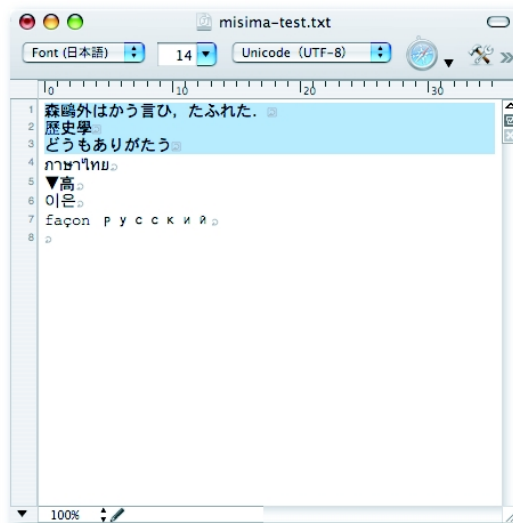


図 36 Jedit X 変換後

TeX 変換オプション付きの変換前後を、それぞれ図 37 (40 頁), 図 38 (40 頁) に示す。

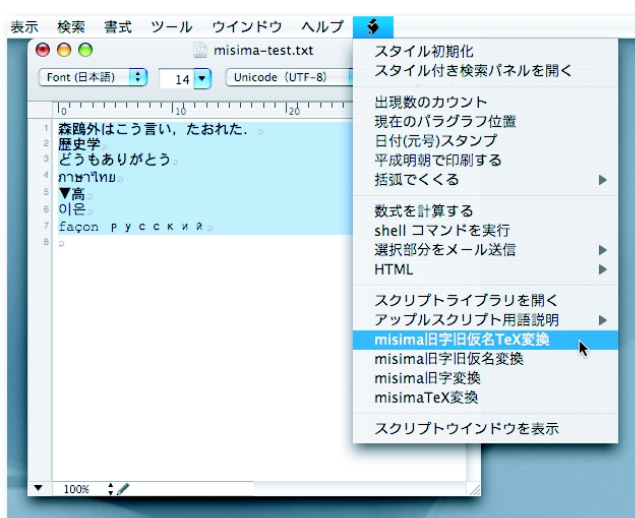


図 37 Jedit X TeX 変換前

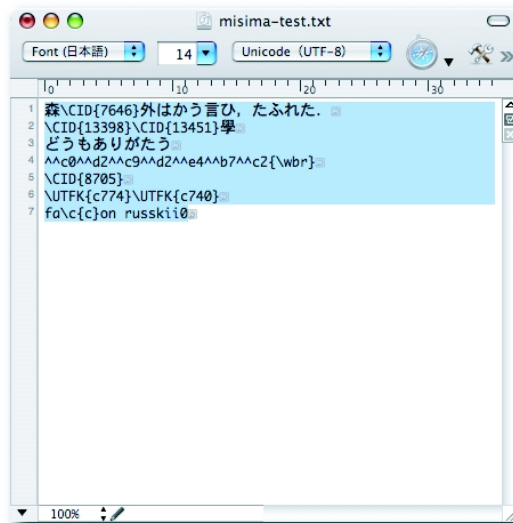


図 38 Jedit X TeX 変換後

4.8 TeXShop エディタ

TeXShop エディタで文書を作成するなかで、misima 旧字・旧仮名遣い変換が必要となるテキストを選択 (マウスで範囲をドラッグ) し、適当な変換オプションにバインドした misima 用 AppleScript を実行すればよい。

スクリプトを実行すると、選択テキストが変換結果で置換される。

テキストが選択されていないと、エラーメッセージを出力する。

変換対象テキストの文字数は1回あたり8,000バイトまで可能である。

もし、結果が変わるはずなのに変換されない場合(時折発生する)、再度変換操作を行ってみるとよい。万一、変換中になんらかの障害が発生してテキストが失われるような事態が起きたら、「Command+Z」でアンドゥーして戻してほしい。

変換前後をそれぞれ図 39 (41 頁), 図 40 (41 頁) に示す。

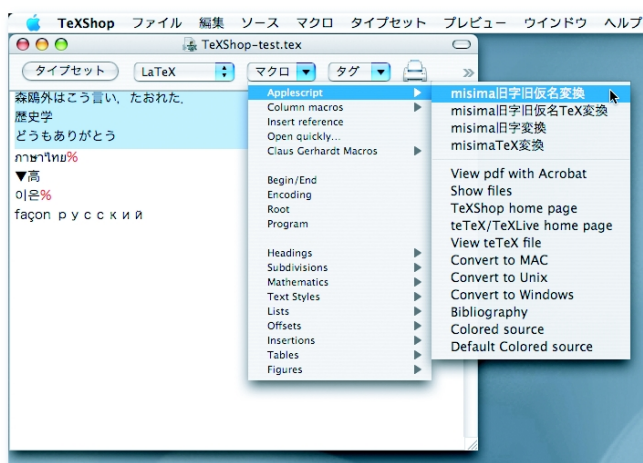


図 39 TeXShop 変換前

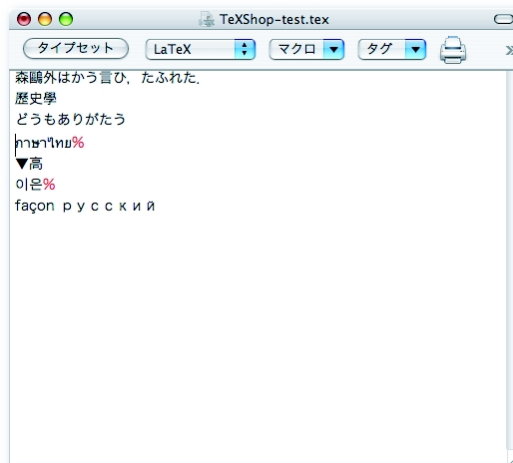


図 40 TeXShop 変換後

TeX 変換オプション付きの変換前後を、それぞれ図 41 (41 頁), 図 42 (41 頁) に示す。

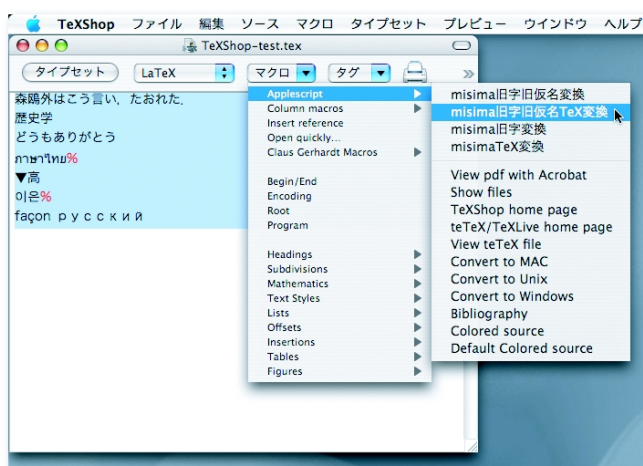


図 41 TeXShop TeX 変換前

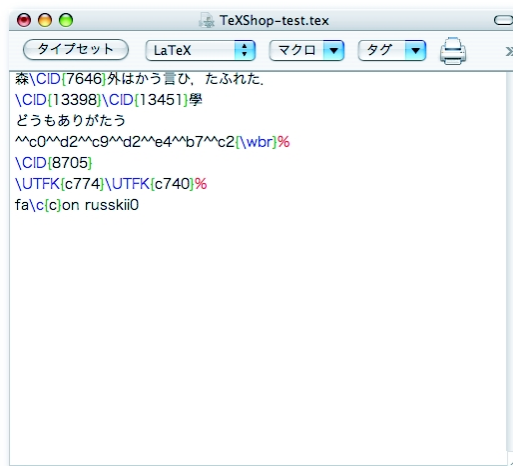


図 42 TeXShop TeX 変換後

表 13 misima オプション引数

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
-h	十六進数形式変換を行う
-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	T _E X OTF package UTF-8 数値参照
a	T _E X OTF package CID 参照
m	T _E X 今昔文字鏡大漢和文字番号参照
-x	引数により T _E X 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 (中文繁體字)
-m	マーキングを行う
h	HTML タグ出力
t	T _E X 命令出力
-g	ドイツ語 7 ビット形式で T _E X 変換を行う
-b	BibT _E X 対応形式で T _E X 変換を行う
-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
-u	misima SOAP サーバ・エンドポイント URL を指定する
URI	misima SOAP サーバ URI (省略時は筆者サーバ)

表 14 Emacs コマンド一覧

コマンド	misima オプション
<code>misimasoap-region</code>	<code>= -kyit -s c</code>
<code>misimasoap-tex-region</code>	<code>= -kyit -s a -x a</code>
<code>misimasoap-tex-germany-region</code>	<code>= -kyitg -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^{*1}</code>
<code>misimasoap-kyuji-buffer</code>	<code>= -t -s c^{*1}</code>
<code>misimasoap-tex-buffer</code>	<code>= -kyit -s a -x a^{*1}</code>
<code>misimasoap-kyuji-tex-buffer</code>	<code>= -t -s a -x u^{*1}</code>
<code>misimasoap-create-buffer</code>	<code>= -kyit -s c^{*2}</code>
<code>misimasoap-tex-create-buffer</code>	<code>= -kyit -s a -x a^{*2}</code>

*1 当該バッファ

*2 新規バッファ

5 misima 処理方式

5.1 入力テキスト形式

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ソースについても同様である。

ロシア語等外国語は、茶筌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、茶筌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。

5.2 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。

5.3 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. $\text{T}_{\text{E}}\text{X}$ 漢文訓点変換
2. 用字・用語変換
3. 旧仮名変換
4. $\text{T}_{\text{E}}\text{X}$ 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. $\text{T}_{\text{E}}\text{X}$ 多言語文字変換
9. 仮名反転変換
10. 十六進数形式変換

6 misima 辞書

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 クライアントのうちユーザ辞書を指定できるのは Web ブラウザのみである。misima SOAP クライアントでは SOAP サーバが SOAP 用ユーザ辞書を読み込む設定になっている。SOAP 用ユーザ辞書は康熙字典別掲漢字・人名漢字を変換しない指定を行ったものである。これは Windows アプリケーションではこれらの漢字を表示できないものが多いためである。

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。

以下、各種変換定義仕様について説明する。なお、misima サーバはユーザ独自の TeX 変換テーブルを追加する機能を有しているが、misima SOAP ではサポートしていないので説明は省略する。

6.1 旧字変換定義

新字とこれに対する旧字を定義する。新字-旧字の対応は一對一である。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 (6.3 節・48 頁) または単純変換定義 (6.4 節・49 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、??新字-旧字対応表 (??頁) を参照。

旧字変換の結果は、TeX 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」(二点付きしんによりの「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 43 (45 頁) に、各定義内容を表 15 (46 頁) に示す。

```
s 新字 旧字#FFFF#99999#999999
```

図 43 旧字変換定義

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 TeX 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。TeX 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。TeX 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 TeX 今昔文字鏡番号変換指定を行ったとし

表 15 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

でも、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 `悅` に変換する。
- 変換指定が $\text{T}_\text{E}\text{X}$ 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 $\text{T}_\text{E}\text{X}$ 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 $\text{T}_\text{E}\text{X}$ 今昔文字鏡番号参照 (`-s m`) のとき、悦 を `\MO{010629}` に変換する。

6.2 旧仮名変換定義

misima 辞書旧仮名変換定義は主に、本則変換 (7.2 節・51 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)」「しょうとう(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 44 (47 頁) に、各定義内容を表 16 (47 頁) に示す。

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、

k 9 新仮名 旧仮名

図 44 旧仮名変換定義

表 16 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）

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 17（47 頁）に品詞コードの一覧を示す．

表 17 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

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．

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

6.3 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。定義は文献[4]を参考にした。

用字・用語変換は、その目的から、旧仮名遣い変換及び旧字変換とセットで用いるべき補足機能である。

用字・用語変換定義のフォーマットを図 45 (48 頁) に、各定義内容を表 18 (48 頁) に示す。

y 変換前 変換後

図 45 用字・用語変換定義

表 18 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

用字・用語変換処理は、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、対象テキスト(語)の **変換前** を **変換後** で置換する。ヒットしないとき、なにもしない。

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。

用字・用語変換の後に旧字変換を独立して行うので、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(この2語はシステム辞書に登録してある)。

用字・用語変換定義の例を次にあげる。

y 画期的 劃期的
y 兄嫁 嫂
y エジプト 埃及

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。

6.4 単純変換定義

単純変換定義のフォーマットを図 46 (49 頁) に、各定義内容を表 19 (49 頁) に示す。

t 変換前 変換後

図 46 単純変換定義

表 19 単純変換定義内容

定義	説明
t	固定
変換前	変換対象語
変換後	変換後

単純変換処理は、単純変換定義に従って、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。

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、単純変換では茶筌の語分割結果、語境界に依存せず置換する。

マーキングの対象外である。

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及び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。

例を以下にあげる。

t 缺伸 欠伸

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。

t 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．

t 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．

t 々

同ノ字点「々」を「 」（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）．

6.5 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（5.3 節・44 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「あゐ」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．

7 旧仮名遣い変換仕様

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．

7.1 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 20 (51 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 20 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。

7.2 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 21 (53 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。

7.3 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 22 (54 頁) 変換を行う。

7.4 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。

8 繰返し符号変換仕様

繰返し符号変換処理は以下の基準に従う。

- 半濁音（ばびぶべぼパピプペポ）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「ゝ」（「ぢゝ」）とする。仮に一語内に「ぢぢ」を有する場合も「ぢゝ」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（「立てゝ」）
 - * 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（「することゝする」）「～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（「言葉でゝも」）「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（「をのゝ刃」）「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。

9 T_EX 変換仕様

9.1 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`^{*12} と同一である。

9.1.1 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。^{*13} Babel パッケージ、OTF パッケージの利用を想定している。

^{*12} 「UTF-8 による TeX 文書の作成」<http://yasuda.homeip.net/tex/utf82tex.html> を参照。

^{*13} 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。

表 21 旧仮名本則変換

文字	変換仕様
わ	<ul style="list-style-type: none"> (1) 語中・語末では「は」に置換する．語頭では置換しない． (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.
い	<ul style="list-style-type: none"> (1) 語中・語末では「ひ」に置換する．語頭では置換しない． (2) 形容詞 のとき語末の「い」は置換しない． (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」 (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」 (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」 (6) 助動詞 / 形容詞・イ段 のとき「い」は置換しない． (7) 助動詞 特殊・ナイ のとき「い」は置換しない． (8) 助動詞 特殊・タイ のとき「い」は置換しない． (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. (10) 「ゐ」に変換すべき語は辞書に登録されるものとする．
う	<ul style="list-style-type: none"> (1) 語中・語末では「ふ」に置換する．語頭では置換しない． (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」 (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． (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． (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.
え	<ul style="list-style-type: none"> (1) 語中・語末では「へ」に置換する．語頭では置換しない． (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. (3) 「ゑ」に変換すべき語は辞書に登録されるものとする．
お	<ul style="list-style-type: none"> (1) 語中・語末では「ほ」に置換する．語頭では置換しない． (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. (3) 「を」に変換すべき語は辞書に登録されるものとする．
じ	<ul style="list-style-type: none"> (1) 「ぢ」に置換する． (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. (3) 「すさまじい(形容詞・イ段)」型の「じ」は「ぢ」としない． (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．
ず	<ul style="list-style-type: none"> (1) 「づ」に置換する． (2) 「感ずる(サ変・-ズル)」の「ず」は「づ」としない (3) 助動詞 特殊・ヌ の「ず」は「づ」としない． (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．
のう	<ul style="list-style-type: none"> (1) 「せつ[のう] [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．
とう	<ul style="list-style-type: none"> (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．

表 22 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	⇒ 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.

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。`-g` オプションを付加するとドイツ語 7 ビット形式で出力する。すなわち、アクセント文字の変換において `{ }` (外側のブレース) を付けず、ウムラウト付文字 `ÄäÛüÖö`, `ß` を、それぞれ `"A" "a" "U" "u" "O" "o"`, `"s"` で出力する。`-b` オプションを付加すると BibTeX でも使用可能な形式で出力する。すなわちラテンアクセント付き文字の変換結果において外側をブレースで囲んで出力する。例えば、`é` を `{\'e}` に変換する。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^{*14}、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。^{*15}

キリル文字言語

Unicode Cyrillic U+0400–U+04FF^{*16} 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 (9.1.2 節・56 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. Слепужин 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>` タグ範囲内でない教会スラヴ語・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 23 (55 頁) で両者を比較せよ。

表 23 教会スラヴ語変換比較

記法	記述	出力	\TeX 出力
\LaTeX 記法	<code><misima_ocs>А</misima_ocs></code>	<code>\OCSYUSM{}</code>	А
T2D	А	<code>\CYRLYUS{}</code>	А

ギリシア語

^{*14} 1992 年のユーゴスラビア解体によって名称は消滅。

^{*15} 文献 [14, pp.327–8] を参照。

^{*16} <http://www.unicode.org/charts/PDF/U0400.pdf>

Unicode Greek U+0370–U+03FF^{*17} (ただしコプト語は除く) 及び Greek Extended U+1F00–U+1FFF^{*18} 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{UCS2 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, ハングルが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{UCS2 コード}` (中国語簡体字), `x:t` のとき `\UTFT{UCS2 コード}` (中国語繁体字), `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 (9.1.2 節・56 頁) を用いる。

特殊異体字

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。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。 は削除される。仮に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。登録文字を表 24 (57 頁) に示す。

\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。

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

拡張変換

ヴェトナム語, グルジア語テキストをそれぞれ $\text{Vn}\TeX$, `mxdreuli` パッケージで処理できる形式に変換する。また機種依存文字^{*19}を OTF パッケージの命令に変換する。「」「」「」「」「」「」などが

^{*17} <http://www.unicode.org/charts/PDF/U0370.pdf>

^{*18} <http://www.unicode.org/charts/PDF/U1F00.pdf>

^{*19} 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。

表 24 特殊異体字登録文字

入力	出力	組版結果
吉	\CID{13706}	吉
高	\CID{8705}	高
崎	\CID{14290}	崎
碯	\CID{17009}	璣
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

処理できる .[-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 pTeX 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 pTeX 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]

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]

9.1.3 T_EX 変換例

本節では T_EX 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_EX 原稿を UTF-8 で準備し, これを misima により変換し, p_LT_EX 2_ε でタイプセットしたものである。コマンドライン手順は Windows の場合, 図 47 (58 頁) のとおりである。なお, 例のタイプセットのために T_EX 原稿のプリアンブルには, 図 48 (59 頁) に示す内容を指定した。

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

図 47 misima 変換, p_LT_EX タイプセット手順

ラテン文字言語について, 図 49 (60 頁) に misima への入力を, 図 50 (61 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 51 (62 頁), 組版結果: 図 52 (63 頁) に示す。

古典ギリシア語例を入力: 図 53 (63 頁), 組版結果: 図 54 (64 頁) に示す。

タイ語例を入力: 図 55 (64 頁), 組版結果: 図 56 (64 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 57 (65 頁), 組版結果: 図 58 (65 頁) に示す。

CJK 統合漢字 (日本語 / 中国語 / 韓国語) 例を入力: 図 59 (66 頁), 組版結果: 図 60 (67 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 (拙作: 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 48 文書例プリアンプル

```
\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 49 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»

8 juillet 2008

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«

8. Juli 2008

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.

Czechoslovak \TeX Users Group

8. července 2008

図 50 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
велета. 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'ятвою <ил'и <луч'єніємъ <архїер'ейскимъ <ил'и <їер'ейскимъ б'ысть, <ил'и
''аще кл'ятву <отц'а своег'w <ил'и м'атере свое'я навед'е на с'я, <ил'и своем'у прокл'ятію
подпад'е, <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 51 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»

8 июля 2008 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/>

8 липня 2008 р.

教會スラヴ語 ᄀᄂᄆᄋᄎ

Їще же под' клѣтвомъ ѿнѣ ѿбѣненіемъ архіерейскимъ ѿнѣ іерейскимъ бысть, ѿнѣ ѿще клѣтвѣ ᄆᄆᄆᄆᄆᄆ ᄆᄆᄆᄆᄆᄆ ѿнѣ мѣтерѣ своѣѣ навѣдѣ на сѣ, ѿнѣ своѣмъ проклѣтїю подпадѣ, ѿнѣ клѣтвѣ прѣсвѣтлѣ, ѿнѣ ѿнѣмнѣ нѣкїмнѣ грѣхѣ ѿкв человекѣкѣ свѣзѣа:

ᄀᄂᄆᄋᄎ sample

8 июля 2008 г.

图 52 キリル文字言語組版出力例

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

图 53 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 ἐτάιρων.
 ἀλλ' οὐδ' ὣς 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, εἶπε καὶ ἧμῖν.

[Ὅμηρος]

图 54 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%
เป็นมนุษย์สุดประเสริฐเลิศคุณค่ากว่าบรรดาฝูงสัตว์เดรัจฉานจงฝ่าฟันพัฒนาวิชาการอย่าล้างผลาญฤๅเช่นฆ่าบีฑา
ใครไม่ถือโทษโกรธแข่งชดชืดชืดต่ำหัดอภัยเหมือนกีฬาอัชฌาษัยปฏิบัติประพฤติกฏกำหนดใจพูดจาให้จ๊ะๆ จ๋าๆ น่าฟัง
เอ๋ยฯ
\hfill\today
```

图 55 タイ語入力例

タイ語 ภาษาไทย

เป็นมนุษย์สุดประเสริฐเลิศคุณค่ากว่าบรรดาฝูงสัตว์เดรัจฉานจงฝ่าฟันพัฒนาวิชาการอย่าล้างผลาญฤๅเช่นฆ่าบีฑา
 ใครไม่ถือโทษโกรธแข่งชดชืดชืดต่ำหัดอภัยเหมือนกีฬาอัชฌาษัยปฏิบัติประพฤติกฏกำหนดใจพูดจาให้จ๊ะๆ จ๋าๆ น่าฟัง
 เอ๋ยฯ

8 กรกฎาคม พ.ศ. 2554

图 56 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 57 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 8 tháng 7 năm 2008

図 58 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}
FAQ
FAQ
“ FAQ 가”

\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。
有痼新结构的细节、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。
%%</misima_cjk>

%%<misima_cjk_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>

```

図 59 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

FAQ

“ FAQ ” “ FAQ 가”

骨練平。

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。

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 60 CJK 統合漢字組版出力例

9.2 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 $\text{T}_\text{E}\text{X}$ 漢文訓点組版スタイル `sfkanbun.sty` に準拠したフォーマットに、ベタの文字列を整形する。p $\text{T}_\text{E}\text{X}$ 縦組で使用する。必要最低限の入力で訓点付漢文を組むことができるよう工夫したつもりである。

9.2.1 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 [9.2.2 節・68 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 61 (68 頁) に示す。

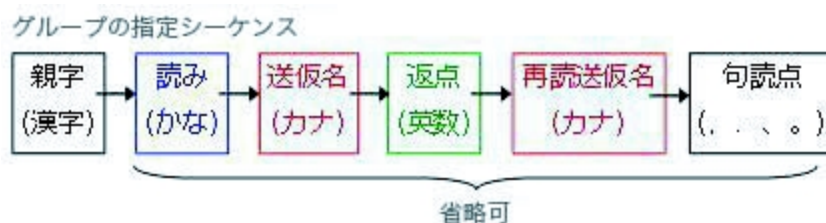


図 61 グループ指定シーケンス

変換例を図 62 (69 頁) に示す。また、変換におけるグループの概念を図 63 (69 頁) に示す。

9.2.2 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、 $\text{T}_\text{E}\text{X}$ 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、 $\text{T}_\text{E}\text{X}$ 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。 $\text{T}_\text{E}\text{X}$ 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 ($\text{T}_\text{E}\text{X}$ 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 62 漢文訓点変換例

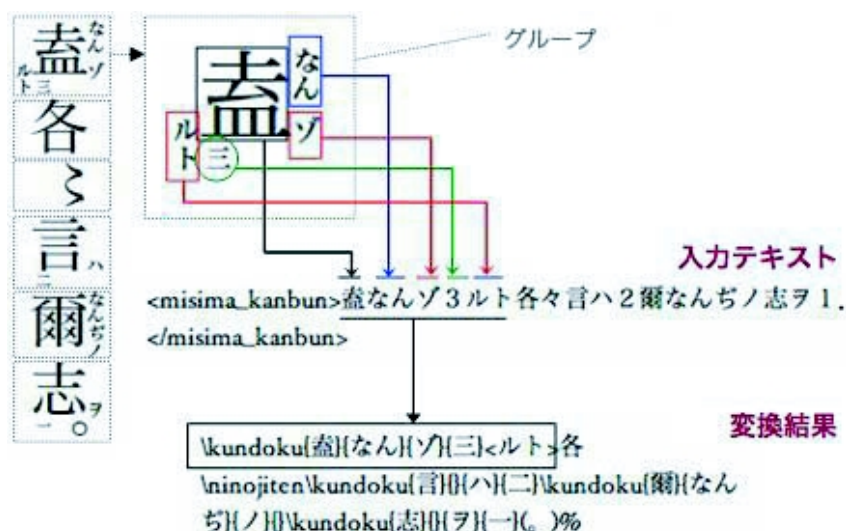


図 63 漢文訓点変換イメージ

} 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，音符号 に変換する。音符号 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。

9.2.3 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 64 (71 頁)、図 65 (72 頁) に示す。

9.3 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」(噛め)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」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 66 (72 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天下之英オヲ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，
人尤もっとも謂いヒ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1R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 64 漢文訓点変換例 (入力)

10 十六進数形式変換仕様

十六進数形式変換機能は、AppleScript SOAP インタフェースなど、Unicode 文字を扱えない処理系を通過させる必要のある場合のために支援している。

AppleScript `call soap` 命令は通常の Unicode 文字を上位に返却するときエスケープエラーを起こす。十六進数形式で Unicode 文字を処理することにより、この制限を掻い潜ることができ、最終アプリケーション

於_レ是_ニ許_レ之_ヲ。
 闔_レ廬_ニ曰_ク、「可_シ試_ミ以_テ婦_人乎_ヲ」。
 曰_ク、「可_{ナリ}」。
 得_レ路_ヲ。
 人_ヲ隨_ニ其_ノ住_ニ、尋_ニ向_ニ所_ニ、遂_ニ迷_ニ不_レ復_ル。
 郡_下一_ノ詣_ニ、太_守一_ノ說_ク如_レ此_ヲ、太_守即_チ遣_下之_ヲ。
 既_ニ出_テ得_ニ其_ノ船_ヲ、便_チ扶_ニ向_ニ路_ニ、處_ニ處_ニ誌_レ之_ヲ、及_ニ之_ヲ。
 聞_ク丙_ノ吏_ノ部_ノ有_テ以_テ博_ク學_ク宏_ク辭_ク、選_ブ者_上、人_尤謂_ニ之_ヲ。
 才_ト得_テ美_シ仕_甲。
 樂_シ得_テ天_下之_ヲ英_才而_チ教_育之_ヲ。
 顏_淵・季_路待_子曰_ク、盍_ル各_々言_ハ爾_志。
 志_燈。

図 65 漢文訓点変換例（出力）

そろそろ
 それぞれ
 散り散り
 散り
 それ
 そろ

図 66 くの子点変換

ンで Unicode 文字に復元する可能性を与える。

例えば、次の短い misima クライアントプログラム (Perl) は、Perl `pack` 関数のオペランドに misima 十六進数形式出力を埋め込むことによって Unicode 文字に復元し、その結果を画面に出力する。この実行結果 (Mac OS X Tiger) を図 67 (73 頁) に示す。

```

#!/usr/bin/perl
use utf8;
binmode STDOUT, ":utf8";

my $MISIMA = "/usr/bin/java misima.misimaSoapClient";
my $ECHO = "/bin/echo";
my $OPTS = "-h -kiytq -s c"; # -h: 十六進数形式変換指定
my $misimaInput
    = "森鷗外はこう言い、たおれた。\\n 内田百間は団扇であおいだ。\\n";
my $misimaResult
    = ` $ECHO -n "$misimaInput" | $MISIMA $OPTS `;

print "Input:\\n"; print $misimaInput; # misima 入力
print "Result:\\n"; print $misimaResult; # misima 出力 (十六進形式)
print "Convert:\\n";
# 十六進形式出力を通常の Unicode 文字に復元
foreach my $line (split(/\\n/, $misimaResult)) {

```



```

my $command = "print pack(\"U*\", $line);";
eval $command;
}

```

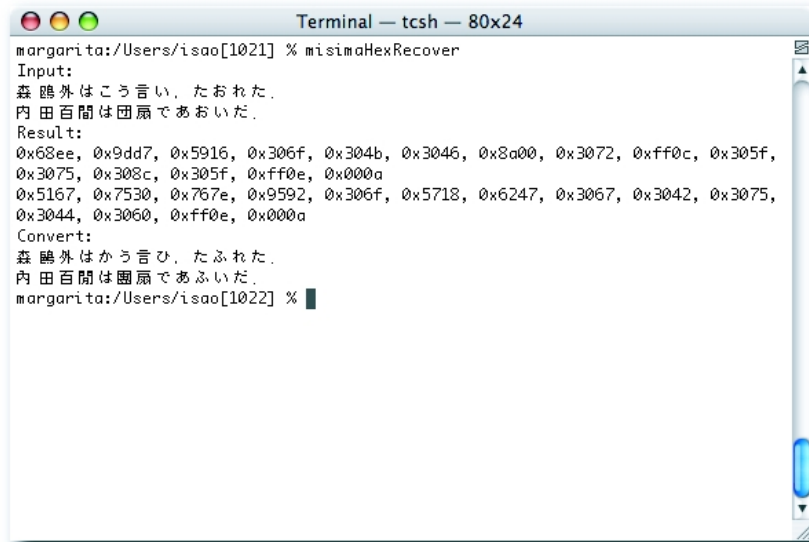


図 67 十六進形式出力 / Unicode 文字復元

11 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。重要な文書に対しご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。

11.1 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、必要ならユーザー辞書に追加する。

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）。

11.2 字音仮名遣い要因

本システムは字音仮名遣いの正確な変換ができない。misima には漢語は漢字で入力するのがよい。

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「しょう(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ため、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。

11.3 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。

11.4 文脈要因

本システムは文脈を判断できない。

筆者の思うに、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。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すると「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かうわけであるが、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。

11.5 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、辞書にエントリがないため茶筌が解析できず、不当な語分割を齎し、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。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。

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。

12 misima SOAP プログラミング

misima SOAP サーバ `misimaSoapConnector` の SOAP インタフェースに準拠すれば利用者独自にクライアントプログラムを作成することができる。本節は Java アプリケーション, `Elisp`, `秀丸エディタ`, `Microsoft Office`, `AppleScript` の各テーマごとに作成方法概略を示す。

なお, 利用者独自に作成したクライアントで筆者サイトの misima SOAP サーバにアクセスする場合は, 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。

12.1 Java アプリケーション

本節では, misima クラスライブラリを用いた misima SOAP Java クライアント作成の流れを説明する。misima クラスライブラリの API ドキュメントは `build/api/` にある。misima SOAP Java クライアントは次の 4 段階を実装する。

オプション解析

指定した misima オプションを `misimaParameter` クラスで解析する。インスタンスを生成した段階でオプションチェックが行われオブジェクトに格納される。misima 変換指定とサーバ URL を `misimaParameter` のメソッドで取得する。misima オプションについては表 13 (42 頁) を参照。

変換対象テキスト取得

変換対象テキストを文字列変数に格納する。misima SOAP クラスライブラリでは, 標準入力 UTF-8 テキストを文字列変数に蓄積する `misimaReader` クラス, クリップボードから入力する `misimaClipboard` クラスをサポートしている。

SOAP 変換

SOAP 接続 `misimaSoap` クラスのインスタンスを生成し, misima SOAP 変換を実行する。

変換結果テキスト出力

変換結果を編集出力する。misima SOAP クラスライブラリでは, UTF-8 テキストを標準出力に書き出す `misimaWriter` クラス, クリップボードに出力する `misimaClipboard` クラスをサポートしている。

変換対象テキスト取得と変換結果テキスト出力はクライアントアプリケーションの形態により千差万別であろう。一方, オプション解析と SOAP 変換はどんな実装にも必要になる部分である。

標準入力から対象テキストを読み, 結果を標準出力に書くサンプルは次のようなものである。

```
// misima パッケージのインポート
import misima.*;
//... 省略
/**
 * 1.misima オプションを解析
 */
```

```

// misima オプション (arg) をセットして misimaParameter を生成する .
misimaParameter mp = new misimaParameter(args);
// misima オプション
String param1 = mp.getParameter();
// misima SOAP サーバ URL
String endpoint = mp.getEndpoint();

/**
 * 2. 変換対象テキストの準備 (標準入力)
 */
// 標準入力から UTF-8 テキストを読むリーダオブジェクトを生成する .
misimaReader mr = new misimaReader();
// param2: 変換対象テキスト
String param2;
try {
    // read() メソッドで文字列に格納する .
    param2 = mr.read();
}
catch (IOException e) {
    System.err.println(e.toString());
}

/**
 * 3.SOAP コール
 */
// misimaSoap (SOAP 接続) オブジェクトを生成する .
misimaSoap ms = new misimaSoap(param1, param2, endpoint);
// misima SOAP サーバで変換し, 結果を convert に格納する .
String convert = ms.misimaSoapTransaction();
// 変換結果が null の時, 入力を戻す .
if (convert == null) {
    result = param2;
}
// 変換結果をセットする .
else {
    result = convert;
}

/**

```

```

* 4. 変換結果の出力 (標準出力)
*/
// 標準出力に UTF-8 変換結果を書くライタオブジェクトを生成する .
misimaWriter mw = new misimaWriter();
try {
    // write() メソッドで文字列を出力する .
    mw.write(result);
} catch (IOException e) {
    System.err.println(e.toString());
}

```

クリップボードから対象テキストを読み、変換結果を再度クリップボードにストアする場合、変換対象テキスト取得と変換結果テキスト出力は次のとおりである。

```

/**
* 2. 変換対象テキストの準備 (クリップボード)
*/
// クリップボードから対象テキストを読み、送信テキストにセットする .
misimaClipboard mcl = new misimaClipboard();
String param2 = mcl.getmisimaClipboard();
//... 省略
/**
* 4. 変換結果の出力 (クリップボード)
*/
// 変換後テキストをクリップボードに書く .
mcl.setmisimaClipboard(result);

```

上記 misima クラスライブラリを用いず、WSDL (Web Services Description Language) をもとにプログラムを書くことも可能である。Axis ユーティリティ [WSDL2Java](#) を用いて WSDL を処理し、SOAP クライアントのための Java スタブクラスコードを自動生成する。これに独自ロジックのコードを加えて行く。Java 言語での SOAP クライアントプログラミングの一般的アプローチである。

WSDL は SOAP サービスインタフェースを記述したものである。[misimaSoapConnector.wsd1](#) を添付している。これは Axis のユーティリティ [Java2WSDL](#) で自動生成したものである。<http://yasuda.homeip.net/axis/services/misimaSoapConnector?wsdl> にアクセスしても misima SOAP の WSDL を取得できる。

12.2 Emacs Lisp

Elisp は外部プログラムをプロセス起動する関数 `call-process-region` をサポートしている．これを用いて変換目的に沿った `misima` オプションとともに Java クライアントを指定し，標準出力を Emacs バッファに挿入すればよい．コーディングシステムを UTF-8 にセットしておくこと．

以下の例は，テキストリージョンの先頭と末尾をそれぞれ `start` と `end` の変数で受け取り，リージョンテキストを `misima SOAP` 変換結果で置換する関数である．^{*20}

```
;; リージョンを misima SOAP 変換し結果で置換する．
;; 旧仮名/旧字/単純/用字・用語/繰返/TeX 文字/漢文/くノ字変換
(setq misimasoap-path "/home/user/bin/misima.sh")
(set-default-coding-systems 'utf-8)
(defun misimasoap-tex-region (start end)
  "Replace selected region texts"
  (interactive "r")
  (call-process-region ;; リージョンを与えてプロセスを起動
    start end
    misimasoap-path
    t (list t nil) nil
    "-ykit" "-s" "a" "-x" "a") ;; misima オプション
)
```

12.3 秀丸マクロ

秀丸エディタは，外部プログラムを起動し終了をウェイトする `runsync2` 命令をサポートしている．これを利用して `misima SOAP` Java クライアントを起動することにより `misima` 変換を行う．

`runsync2` は選択テキストを標準入力として指定プログラムに渡すことが可能であるが，`misima` のような UTF-8 入出力を行うプログラムの場合，標準入出力を利用すると文字化けが発生してしまう．クリップボードのテキストについては秀丸が Unicode をサポートしているため，クリップボード経由の変換により，この問題が回避できる．すなわち，`misimaSoapCbClient` クラスモジュールを利用して，クリップボードから入力したテキストを `misima SOAP` 変換し，変換結果を再度クリップボードに出力する．その後，選択テキスト領域に戻すマクロをコードすればよい．

例を示す．“ ” は行の折り返しを示す．

^{*20} `call-process-region`，Emacs Lisp プログラミングの詳細については，文献[11]を参照のこと．

```
// misima オプション
$param = "-kyit -s a -x kuitx ";

if (!selecting) {
    message "文字列を選択してください";
    endmacro;
}

// クリップボードに選択文字列を切り出す .
cut;

// misima SOAP サーバに変換要求し、リターンをクリップボードに書く .
runsync2 "misimalaunch javaw misima.misimaSoapCbClient
-u http://yasuda.homeip.net/axis/services/misimaSoapConnector "
+ $param;

if (!result) {
    message "実行に失敗しました";
    // 異常の場合、クリップボードから戻して終了する .
    paste;
    endmacro;
}

// misima 変換結果をペースト
paste;
```

`runsync2` マクロの引数にある `misimalaunch` は、単にそのオペランド指定プログラムをプロセス起動するためだけの `misima SOAP` ツールである。`runsync2 "javaw misima.misimaSoapCbClient..."` でも変換操作は期待通りに働くが、一方で変換中にコマンドプロンプトの画面が出てしまう。`misimalaunch.exe` によればこの画面を抑止できる。

`misima` オプションについては表 13 (42 頁) を参照。

秀丸エディタ以外のアプリケーションでも、マクロプログラミングインタフェースを備えていれば、上記と同様の考え方で `misima SOAP` を組み入れることが可能であると思われる。

12.4 Microsoft Office

Microsoft Office XP Web Services Toolkit 2.0 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。

Web Service References Tool において、「Web サービス URL (U) - URL (R)」に `misima SOAP サービス 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 68 (80 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 (69・80 頁)。

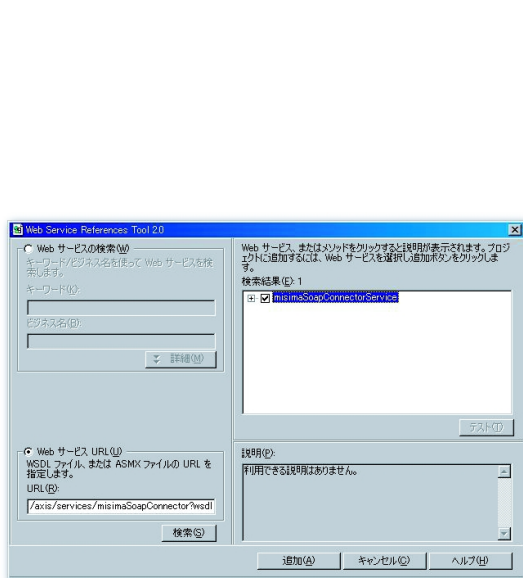


図 68 misima SOAP スタブクラスの追加

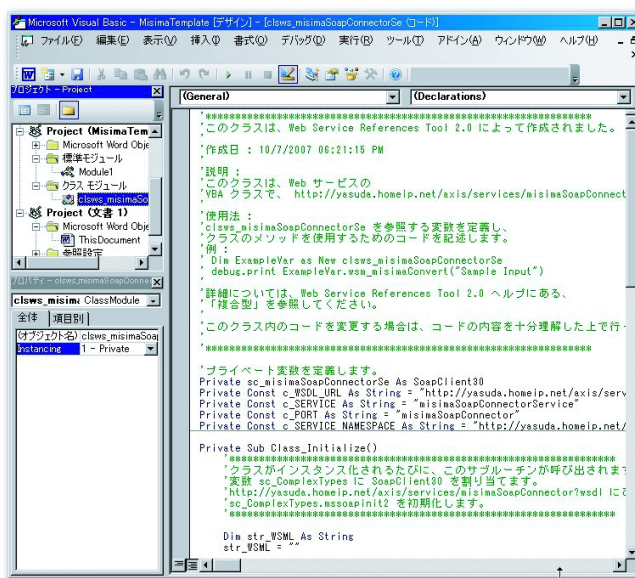


図 69 misima SOAP スタブクラスコード

この自動生成された `clsWS_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。`misima` オプションについては表 13 (42 頁) を参照。エラー処理などの付随コード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 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"
' 選択範囲テキストの取得
sTargetText = Selection.Text
' SOAP misima 変換サーバ呼び出し
sConvertResult = misimaWebService.wsm_misimaConvert( _
    sArgument, sTargetText)
' 変換結果テキストで選択テキスト(元テキスト)を置換
Selection.Text = sConvertResult
End Sub

```

12.5 AppleScript

Mac OS X のアプリケーションは AppleScript によりその動作を制御できるものが多い。

AppleScript は SOAP Web Service をコールする命令(`call soap` 命令)をサポートしており、^{*21} Emacs・秀丸用クライアントとは異なり、misima SOAP Java クラスライブラリを必要とせずクライアント実装が可能である。

`call soap` 命令を用いると利用者で独自に misima 用クライアント AppleScript を追加することも容易であるが、以下の点に留意すべきである。

- AppleScript `call soap` 命令は、筆者が試験した限りでは、Unicode 文字返却値を正しく処理できない。ASCII 文字列でないと `«Unknown Escape»` が出力されてしまう。
- そのため、misima オプションに“-h”(十六進数形式変換)を指定して、Unicode 文字を UCS 十六進数形式で返すよう misima SOAP サーバに指示する。これによって、ASCII 文字だけで `call soap` 命令を通過させる。misima オプションについては表 13 (42 頁)を参照。
- その後、UCS 十六進数形式の変換結果を Unicode 文字に復元する。十六進数文字列を Perl `pack` 関数に埋め込んで、`do script shell` 命令で呼び出す、などの手続きを行えばよい。^{*22}

Jedit X 用 AppleScript 実装例を以下に示す。コード中の `↵` は行の折り返しであり、これを入力せず前後を一行で記述しなければならない。

```

-- グローバル変数
global misimaOpts -- misima オプション
global tString -- Jedit X 上の選択テキスト

```

^{*21} AppleScript における SOAP 及び XML-RPC サポートについては、文献 [13] を参照。

^{*22} AppleScript における Unicode 文字の取り扱いに関する詳細な解説は、文献 [12] を参照。

```
global rText -- 十六進数形式の misima 変換結果
global sText -- 変換後テキスト

-- misima オプションをセット (-h 必須)
set misimaOpts to "-h -kity -s a -x kutix "
```

```

-- 十六進数形式を Unicode に復元するハンドラ .
on misimacnv(str)
  -- paragraph の参照を設定 .
  script list_paragraphs
    property contents : paragraphs of str
  end script
  -- Unicode 文字復元 .
  set rt to ""
  repeat with line_item in contents of list_paragraphs
    -- Perl pack 関数オペランドに十六進数形式をセット .
    set p to "print pack(\"U*\", " ~
      & (get contents of line_item) & ");"
    -- 改行コード (0x000a) を \n\n (改行文字二つ) に置換 . ~
      (do shell script が改行文字を一つ削除するため)
    set p to replace(p, "\n", 0x000a);", "\n\n") . "\"\\n\\n\";"
    -- Perl pack 関数を実行し, 行を連結 .
    set rt to rt & (do shell script "perl -e " & ~
      quoted form of p)
  end repeat
  return rt
end misimacnv

-- テキスト置換ハンドラ .
on replace(src, tg, rp)
  set delmback to AppleScript's text item delimiters
  set AppleScript's text item delimiters to tg
  set tmpList to text items of src
  set AppleScript's text item delimiters to rp
  set cText to tmpList as string
  set AppleScript's text item delimiters to delmback
  return cText
end replace

```

選択テキストの取得や置換は,アプリケーションがサポートしているオブジェクト仕様に依存する. 例えば, 選択テキストオブジェクトは Jedit X においては `selected text` であるが, TeXShop の場合 `content of selection of front document` である.

13 謝辞

misima のそもそもの利用目的は、筆者自身のモチーフとして、 $\text{T}_{\text{E}}\text{X}$ 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 $\text{T}_{\text{E}}\text{X}$ 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 $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ のコミュニティで公開されている OTF パッケージや sfskanbun 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。

misima は、日本語形態素解析ソフトウェア茶筌の解析結果と、『私の國語教室』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ず、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。

ヴェトナム語、グルジア語及び機種依存文字用の $\text{T}_{\text{E}}\text{X}$ 拡張変換テーブルは大友康寛氏^{*23}より提供いただいたものである。大友氏に感謝申し上げます。

14 利用条件

本利用条件は misima 旧仮名遣い・旧字変換支援及び misima SOAP Web Service 2.5 クライアント（以下「本ソフトウェア」）に適用する。

ただし、表 9（18 頁）に示す同梱クラスライブラリは適用範囲外である。

1. 本ソフトウェアの著作権は筆者 isao yasuda, isao@yasuda.homeip.net（以下作者）に帰属するものとする。
2. 個人の私用目的に限り無償にて使用できる。
3. 商用で利用する場合は作者と協議を行うものとする。
4. 本ソフトウェアの利用、そのための準備、及びその利用結果について、いかなる場合においても無保証である。
5. 本ソフトウェアは、内容を改変しない限りにおいて、自由に再配布できる。再配布の場合は、本利用条件を添付しなければならない。
6. 本ソフトウェアを改変することができる。
7. 本ソフトウェアの改変による成果物を第三者に公開する場合、作者と協議するものとする。

^{*23} <http://www.klavis.info/>

8. 本ソフトウェア及びこれが前提としているサーバ機能について、断りなく仕様を変更する場合がある。
9. 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。

misima システム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。

15 更新履歴

- 2004/11/03 Ver.1.0 Web 版公開
- 2005/07/10 Ver.2.0 T_EX 変換サポート (多言語, 漢文訓点)
- 2005/08/01 Ver.2.0 SOAP Web Service 0.1 パッケージ公開
- 2006/01/01 Ver.2.1 T_EX 変換タイ語, 中韓拡張
- 2006/06/10 Ver.2.2 T_EX 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・ましょ」型旧仮名変換問題訂正
- 2007/08/19 Ver.2.3e 本則「う」変換, T_EX 変換訂正
- 2007/09/23 Ver.2.3f misimaSoapClient 仕様変更, ドキュメント・リリース
- 2007/10/01 Ver.2.3g SOAP 問題訂正, SOAP Web Service 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
- 2007/11/02 Ver.2.4 十六進形式変換, Jedit X, T_EXShop SOAP クライアント
- 2008/01/01 Ver.2.5 T_EX 変換ドイツ語, BibT_EX オプション
- 2008/07/08 Ver.2.5 Web Servlet 版, AppleScript ディスクイメージ追記

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 (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 (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 『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] Chappell D. A., Jewel T. 『Java Web サービス』 長瀬嘉秀監訳, テクノロジックアート訳, オライリー・ジャパン, 2002.
- [9] Graham S., Simeonov S., Boubez T., Davis D., Daniels G., Nakamura Y., Neyama R. 『Java による Web サービス構築』 橋本正, 西本進, 野間克司, 福原信貴監訳, ソフトウェアバンクパブリッシング, 2002.
- [10] 奥村晴彦 『L^AT_EX 2_ε 美文書作成入門』 改訂第 4 版, 技術評論社, 2007.
- [11] Lewis B., LaLiberte D., Stallman R., and the GNU Manual Group 『Emacs Lisp リファレンスマニュアル』 大木敦雄訳, アスキー出版局, 2000.
- [12] Neuburg M., *AppleScript, The Definitive Guide*, 2nd Edition, O'Reilly & Associates Inc., 2006.
- [13] Apple Inc., *XML-PRC and SOAP Programming Guide*, <http://developer.apple.com/documentation/AppleScript/Conceptual/soapXMLRPC/soapXMLRPC.pdf>, 2005.3.
- [14] 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 『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