

る例もあったが、薬名を製剤名で記載し、分量は製剤量を記載することを基本とする。例外的に、分量を原薬量で記載した場合には、必ず【原薬量】と明示する。

4)「用法・用量」における服用回数・服用のタイミングについては、「×3」、「3×」等の情報伝達エラーを惹起する可能性のある紛らわしい表現を、「1日3回朝昼夕食後」のように日本語で明確に記載することにより、紛らわしい記載を速やかに是正する。なお、当分の間、1回量と1日量を併記する場合には、「分3」等の1日量を前提とした表現も許容する。

5)「用法・用量」における服用回数・服用のタイミングについては、医療情報システムにおいて用いられる1回量による処方前提とした標準用法マスタを作成し公表を行う。

6)入院患者用の薬剤を調剤する際に、賦形を行った場合には、薬剤師が、与薬する看護師等に対し、賦形後の調剤量及び1回量を明確に伝達する必要がある。過去の大会において、これらの短期的方策については、本シンポジウムで発表を行っているベンダーにおける病院情報システムについては基本的に対応ができており、その多くは標準パッケージに含まれているとの報告がなされてきた。ただし、レセコンにおける状況は必ずしも明確でないことから、今回は各ベンダーがレセコンを有する場合にはそれらの開発状況についても報告を行うことを要請している。

また、日本薬剤師会と日本病院薬剤師会では標準用法用語集が作成され、これに対応した形で日本医療情報学会によってそれらは標準用法マスタとして公表されているが、本年標準用語集のV2が示されたことからそれについては大江班のメンバーより報告が行われる。

尚、2)にある1回量と1日量を処方せんに併記することは、検討会によって新たに作られた対応策ではなく、現行の保険制度での要求事項であるにも拘わらず、殆ど守られていないことから、短期的方策として、現行の保険制度での記載方法を遵守することが明記されるものであることを再認識する必要がある。

### 3. 長期的方策の実現状況

一方、長期的方策については、報告書に以下のような記載がある。

1)「分量」については、処方オーダーリングシステム等の処方入力画面において、1回量を基本とした入力方法に対応できる処方入力画面を装備し、かつ1回量と1日量についても同一画面で確認できることとする。また、処方オーダーリングシステム等には、原則として服用回数・服用のタイミングに関する標準用法マスタを使用することとする。

2)投薬実施記録としての院内看護システムにおいては、処方せんによる投薬指示が患者に確実に実施されるために、最小基本単位である1回量を基本単位とすることを推進する。

3)調剤薬局において処方内容を再入力することによる情報伝達エラーを防止するとともに、院外処方せんの利便性の向上に資するような、情報技術等(二次元バーコード等)の導入について検討する。

長期的方策である1回量を基本として入力方法に対応したシステムは複数のベンダーにより開発され、大学病院をはじめ一般病院においても導入されているこ

とが前回の連合大会において報告がなされた。そこで本年は各ベンダーに開発状況のみならず、稼働実績についても報告をしてもらう予定である。

また、これらの方式を導入した病院における移行時の課題については、ヒアリング調査を行い、報告をすることとする。

### 4. 電子処方せんに関する最近の話

電子処方せんについては医療情報ネットワーク基盤検討会において平成16年から検討がなされている。同検討会では、平成20年に「処方せんの電子化について」、平成24年には「処方箋の電子化に向けて」、平成25年には「電子処方箋の実現について」なる報告書を公開している。

一方厚生労働省では昨年9月に電子処方箋の実現に向けた工程表も発表されている。また、今後2年を目前に現在はe-文書法において適応外とされている処方せんについて法改正を行うことにより「電子処方箋を違法ではない」とすることが検討されている。

さらに昨年「電子化した処方箋の標準化様式の整備と運用に関する研究(研究代表者大江和彦)」「処方せんの電子化に伴う情報連携。情報利活用・プライバシー保護のあり方に関する調査研究(研究代表者山本隆一)」の2つの厚労科研が実施されており、電子処方箋の実現に向けて検討が行われている。そこで大江班の協力のもと、「電子処方せんにおける電子データ記述規格の標準化」についても現状報告が行われる。

### 5. おわりに

内服薬処方せんの記載の在り方に関する検討会の報告書に記載されている内容については、確実に進展が見受けられる。同報告書が実施期限を定めていない、また国が費用負担をする形ではなく、各医療機関における情報システムのリプレースに合わせた形での導入となっているため、当初報告書が想定していたよりも2～3年の遅れとなっていることは事実である。

我が国においては医療費抑制策として後発品の使用推進策がとられているが、中医協においては、2年毎に処方箋の記載方法が変更されること、及び24年の診療報酬改定において、「一般名処方」と称した処方箋の記載方法が推奨され、かつ診療報酬上の評価(院外処方箋の場合に処方箋料に2点の加算)が行われたことにより、ベンダーはその対応に追われてしまったことも事実である。

平成26年の診療報酬改定ではDPC病院において後発医薬品が評価指標に加えられたことから、今後は後発品使用が進展すると思われ、処方箋記載との関係は薄れていくことも考えられる。

一方電子処方箋とは別にお薬手帳についての電子化についても様々な議論がなされている。ただ、現状で行われているお薬手帳の電子化は保険薬局における電子化であり、調剤情報の記載中心であるため、患者にとってお薬手帳は本来患者自身が自ら服用(使用)している医薬品についての記録を残すものであり、患者が情報コントロール権を有するにもかかわらずそのような感覚を患者が殆ど有していないという由々しき状況にある。処方せんにせよお薬手帳にせよ電子化を図る際に、紙をベースに各種の検討を行うことは必要ではあるが、電子化をベースに検討を行わないと、電

### 3-G-1 共同企画/3-G-1:共同企画7

子化のメリットが活かされないことになりかねないことが危惧される。

電子処方箋の検討が確実に進展しているが、厚労省のネットワーク基盤検討会における議論の主眼は院外処方箋にある。病院情報システムにおける過去の例を考慮するのであれば、院内処方に対して検討を行うことが現実的ではないかと思われる。そのことが、結果的に内服薬処方せんの記載の在り方に関する検討会の報告書の内容の実現の促進剤となるのではないか

と思われる。

#### 参考文献

- [1] 内服薬処方せんの記載方法の在り方に関する検討会報告書 2010.1. <http://www.mhlw.go.jp/shingi/2010/01/s0129-4.html>.
- [2] 電子処方箋の実現について:医療情報ネットワーク基盤検討会報告書 2013.05. <http://www.mhlw.go.jp/stf/shingi/2r9852000002xok3-att/2r9852000002xonn.pdf>.



■ 概要

トランザクションファイルから変換対象を取得し、HL7CDA フォーマットを最新仕様へ対応させると共に、SS-MIX2 標準化ストレージに格納された処方オーダーデータを自動で検知し、HL7CDA 形式に自動で変換して SS-MIX2 拡張ストレージに格納する

■ フォルダ構成

アプリケーションフォルダはコンピュータの任意の場所に配置できる。

フォルダ・ファイル名	内容
ConverterService	アクセス権設定サービスプログラムのルートフォルダ
bin	実行ファイル用フォルダ
Common.dll	共通処理のクラスファイル
HL7.dll	変換処理のクラスファイル
HL7Conf.dll	監視処理のクラスファイル
HL7Monitoring.exe	監視処理の実行ファイル
log4net.dll	log 出力に関するクラスファイル
log4net.xml	log の設定
TransactionLogAnalysis.exe	変換処理の実行ファイル
TransactionLogAnalysis.exe.config	変換処理の設定
log	ログ出力先フォルダ
conf	設定ファイル用フォルダ
HL7ServiceConf.xml	変換・監視処理の設定ファイル

当該サービスが起動しているかどうかのチェックを、HL7Monitoring.exe を用いて行う。変換処理プロセスの存否を確認し、起動していない場合はプロセスを起動するスクリプトを 10 分に 1 度実行するよう設定する。

■ 設定ファイル記載内容

HL7ServiceConf.xml

設定項目名	内容	設定値
AnalysisFileName	解析開始ファイル名	文字列
AnalysisLineNum	解析開始行	整数
RetryCount	リトライ回数	整数
MaxGoingBackDays	最大取得日数	整数

ExtractionCondition	検索条件	文字列(平文)
▲ DatClassification	データ種別	文字列
▲ ProcessingDivision	処理区分	文字列
▲ MedicalDepartmentCord	診療科コード	文字列
ConvertTargetDate	変換対象日	日付
RootFolderPath	トランザクションログパス	パス文字列

■ 処理内容

1. タスクスケジューラから定時で HL7Monitoring.exe が起動
2. TransactionLogAnalysis.exe の監視スタート
3. TransactionLogAnalysis.exe が停止している場合、起動
4. 変換処理
  - ① トランザクションファイル解析
  - ② 変換対象取得
  - ③ 変換
 ※ 処理に失敗した場合は、設定ファイルのリトライ回数を加算して更新する。

以上.



■ 概要

処理要求ファイルが配置されるフォルダを監視し、ファイルを検知したときフォルダアクセス権の付与または解除を行うプログラムの設計を記述する。

プログラムは Windows サーバ上で動作するサービスである。

■ 必要とするライブラリ等

Windows サービスとして動作させるため、Apache Commons Daemon を使用する。下記を使用する。

- ・ commons-daemon-1.0.15.jar
- ・ prunsvr.exe
- ・ prunmgr.exe

使用ライセンスについては下記を参照のこと。

<http://commons.apache.org/proper/commons-daemon/license.html>

ログ出力には、log4j を使用し、日付でローリングする。

データベースにアクセスするため、下記を使用する。

- ・ commons-dbcp-1.2.1.jar
- ・ commons-dbutils-1.5.jar
- ・ postgresql-9.3-1104.jdbc41.jar

■ フォルダ構成

アプリケーションフォルダはコンピュータの任意の場所に配置できる。

フォルダ・ファイル名	内容
PermissionSettingApps	アクセス権設定サービスプログラムのルートフォルダ
bin	実行ファイル用フォルダ
FolderPermissionControl.exe	サービスの実行ファイル
FolderPermissionControl.jar	サービスのクラスファイル
FolderPermissionControlw.exe	サービスの管理を行う実行ファイル
DeleteOldLog.bat	一定期間経過したログを削除するバッチ
HealthCheck.vbs	サービスの監視&再起動を行うバッチ
conf	設定ファイル用フォルダ
log4j.properties	ログ出力設定ファイル
FolderPermissionControl.properties	動作設定ファイル

log	ログ出力先フォルダ
-----	-----------

当該サービスが起動しているかどうかのチェックを、タスクスケジューラを用いて行う。サービスプロセスの存否を確認し、起動していない場合はサービスを起動するスクリプトを 10 分に 1 度実行するよう設定する。

■ 設定ファイル記載内容

FolderPermissionControl.properties

設定項目名	内容	設定値
ssmix2_root_dir	SS-MIX2 ルートフォルダパス	パス文字列
req_file_dir	フォルダアクセス権設定要求ファイル配置先ディレクトリパス	パス文字列
err_file_dir	処理失敗となった要求ファイルを配置するディレクトリのパス	パス文字列
db_user_name	DB のユーザ名	文字列
db_password	DB のパスワード	文字列(平文)
db_url	接続先 DB 情報	文字列

■ 処理内容

1. 設定ファイル読み込み
2. フォルダアクセス権限編集要求ファイル配置フォルダの監視スタート
3. 要求ファイルの命名規則に一致したファイルを検知
4. 要求ファイル名末尾に.tmp を付与。既に tmp の場合は rty を付与。
  - ① ファイル内容解析
  - ② アクセス権の付与または解除
  - ③ アクセス権一覧ファイルへの書き込み
  - ④ 要求ファイル削除

※ 処理に失敗した場合は、内容をログに出力して削除する。  
 ※ 起動後初回の要求ファイル処理では、前回起動で完了しなかったファイルも処理する。  
 ※ 付与するアクセス権は変更アクセス権とする。  
 ※ 処理対象フォルダが存在しなかった場合、警告ログを出力し、要求ファイルは削除する。

5. DBのアクセス権限一覧に、権限を付与したユーザ・患者ID・オーダ種別を登録する。

■ 要求ファイルフォーマット

● アクセス権付与要求

命名規則 : GRANTyyyyMMddHHmmssfff.req  
 フォーマット : ユーザ名,患者 ID,オーダ種別(要素数3のCSV)  
 各要素はダブルクォート括弧がないものとする。また、CSVの各要素はカンマが含まれる可能性がないことから、カンマエスケープは考慮しない。

例) ファイル名 : GRANT201602222183427456.req  
 内容 : user1,000111234, OMP-01

user1 ユーザに対し、患者 ID000111234 フォルダのオーダ種別 OMP-01 フォルダ配下の全ファイルにアクセス権限を付与する

● アクセス権解除要求

命名規則 : REMOVEyyyyMMddHHmmssfff.req  
 フォーマット : ユーザ名,患者 ID,オーダ種別(要素数3のCSV)  
 アクセス権付与要求と同様、ダブルクォート括弧とカンマエスケープは考慮しない。患者IDとオーダ種別を記載しないことで、ユーザ単位でアクセス権を解除できるものとする。

例 1) ファイル名 : REMOVE201602222183427456.req  
 内容 : user1,000111234, OMP-01

user1 ユーザに対し、患者 ID000111234 フォルダのオーダ種別 OMP-01 フォルダ配下の全ファイルからアクセス権限を解除する

例 2) ファイル名 : REMOVE20160223212249012.req  
 内容 : user1, ,

user1 ユーザに対し、ルートフォルダ以下全てのフォルダに対するアクセス権限を解除する

● 拡張子について

アクセス権付与・解除要求ファイルには状態に応じて下記の拡張子を付与する。

処理中	.tmp
-----	------

処理中にプログラムが終了したファイルの再処理	.rty
------------------------	------

■ アクセス権一覧データベース

DBはPostgreSQL 9.3を使用する。

DB名 : ssmix2pmt

テーブル名 : t\_grant\_info

カラム名	型	サイズ	主キー	内容
user_name	varchar	32	○	ユーザ名
patient_id	varchar	200	○	患者ID
order_kind	varchar	6	○	オーダ種別

■ 初回処理バッチ

設置時、サービスを起動する前にSS-MIX2ルートフォルダに対して下記を行うバッチを手動で実行する。

1. Everyone フルコントロールでフォルダ共有
2. 配下の全ファイル・フォルダのアクセス権を解除
3. 施設で全フォルダにアクセスできるシステム管理者が必要とされるなどの場合は、必要なユーザ・パスに必要な権限を与える
4. アクセス権一覧ファイル(ファイル名の先頭と拡張子で判断)を削除する。

本バッチファイルは、設置時に施設ごとに適切な内容を都度作成する。

以上.

厚生労働科学研究費「電子化した処方箋の標準化様式の整備と運用に関する研究」における調剤レセコン連携に関する開発

**処方箋 CDA\_2 次元シンボル  
処方箋変換プログラム**

株式会社 E M システムズ

◀目次▶

1. 概要	4
2. 機能要件	4
3. 構成	5
フォルダ構成	5
プログラム 入出力関係図	6
処理フロー	6
入出力ファイル	7
3. 画面	8
メイン画面	8
4. 機能	8
処方箋 CDA 追加	8
処方箋 CDA 除外	9
変換実行	10
変換定義・マスタ参照/保存	10
5. 変換仕様	11
処方箋 CDA_2次元シンボル処方箋変換仕様	11
標準用法名称/補足用法名称作成仕様	11
標準用法コードから標準用法名称の作成	11
補足用法コードから補足用法名称の作成	12
6. 定義ファイル	12
変換定義ファイル	12
バージョンレコード変換定義	13
1 医療機関レコード	13
2 医療機関所在地レコード	14
3 医療機関電話レコード	16
4 診療科レコード	17
5 医師レコード	19
11 患者氏名レコード	20
12 患者性別レコード	21
13 患者生年月日レコード	22
14 患者一部負担レコード	23
21 保健種別レコード	24
22 保険者番号レコード	25
23 記号番号レコード	27
24 負担・給付率レコード	28
25 職務上の事由レコード	29
26 老人保健レコード	30
27 第一公費レコード	30
28 第二公費レコード	32
29 第三公費レコード	33
30 特殊公費レコード	34
31 レセプト種別レコード	34
51 処方せん交付年月日レコード	35
52 使用期限年月日レコード	37
61 麻薬施用レコード	38



81 備考レコード.....	38
101 剤形レコード～281 薬品補足レコードについて.....	39
101 剤形レコード.....	42
111 用法レコード.....	44
181 用法補足レコード.....	47
201 薬品レコード.....	49
211 単位変換レコード.....	54
221 不均等レコード.....	54
231 負担区分レコード.....	56
241 1回服用量レコード.....	56
281 薬品補足レコード.....	58
用法名称作成定義ファイル.....	58
出力値定義.....	59
項目値定義.....	60
<b>7. 動作環境.....</b>	<b>61</b>
付録.....	62

## 1. 概要

厚生労働科学研究費「電子化した処方箋の標準化様式の整備と運用に関する研究」調剤レセコン連携に関する開発で、平成 26 年度に確認された処方箋の電子記録様式によって記述された処方情報を調剤薬局で使用されている既存レセプトコンピュータへ取り込めるように電子処方箋データの変換を行うソフトウェアの開発を行う。

## 2. 機能要件

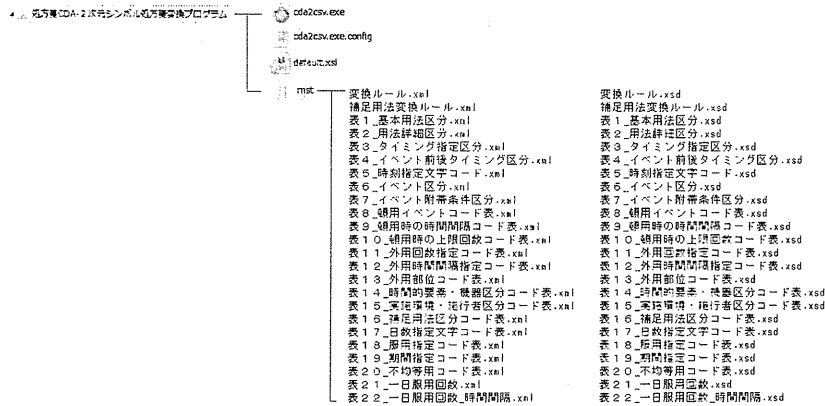
以下の機能を実装する。

- 処方箋の電子記録様式 (HL7CDAR2) から『JAHIS 院外処方せん 2 次元シンボル記録条件規約 Ver1.1』へ変換する機能
- 『処方オーダリングシステム用標準用法「服用回数、服用のタイミングに関する標準用法マスタ」』で定義されている用法コードを解析し、用法名称を生成する機能
- 対象となる処方箋情報ファイル (HL7CDAR2) を指定する機能 (GUI による指定)
- 『JAHIS 院外処方せん 2 次元シンボル記録条件規約 Ver1.1』へ変換したファイルを指定したフォルダに保存する機能 (GUI による指定)
- 処方オーダリングシステム用標準用法「服用回数、服用のタイミングに関する標準用法マスタ」で定義されている表定義情報を参照/変更/保存する機能

### 3. 構成

#### フォルダ構成

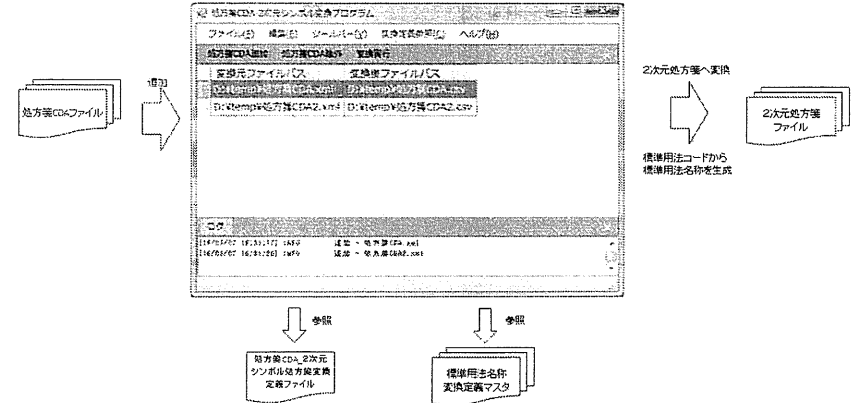
フォルダ構成を以下に記述する。



cda2csv.exe	メインプログラム
cda2csv.exe.config	プログラム設定ファイル
default.xml	処方箋 CDA から 2 次元シンボル処方箋を作成するためのスタイルシート
mst/	標準用法コードまたは補足用法コードから標準用法名称を作成する為の定義ファイルを保存する。
変換ルール.xml	標準用法コードから標準用法名称を作成するための定義ファイル
補足用法変換ルール	補足用法コードから補足用法名称を作成するための定義ファイル
表○○.xml	処方オーダリングシステム用標準用法「服用回数、服用のタイミングに関する標準用法マスタ」で定義されている表ごとにマスタファイル。＊独自の定義を含む
表○○.xsd	対応する xml ファイルの表定義ファイル

#### プログラム 入出力関係図

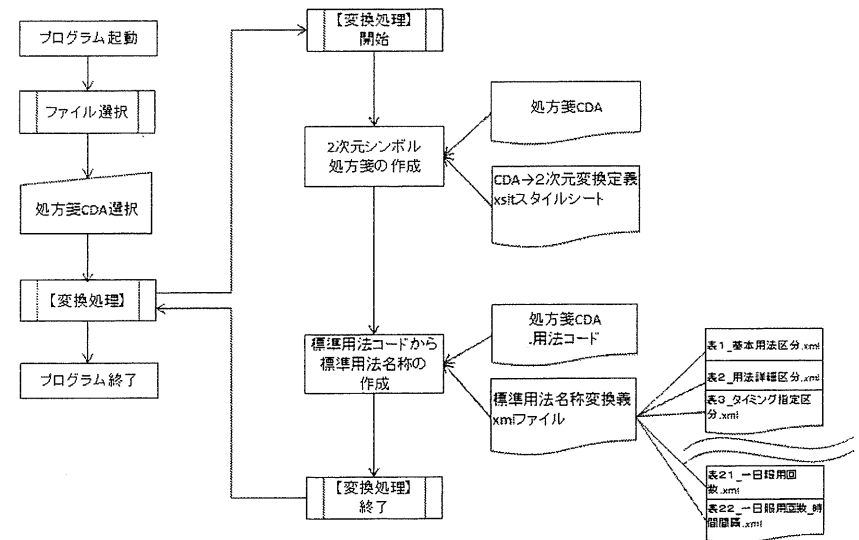
プログラムの入出力を以下に記述する。



各入出力のファイルについては次の項目で説明を行う。

#### 処理フロー

プログラムの処理の流れを以下に記述する。



## 入出力ファイル

入出力ファイルの詳細を以下に記述する。

### 入力ファイル

処方箋 CDA ファイル	GUI で指定された変換対象の処方箋(電子記録様式 (HL7CDAR2) )
--------------	--

### 参照ファイル

処方箋 CDA_2 次元 シンボル処方箋変換定義フ ァイルファイル	変換対象の処方箋から 2 次元シンボル処方箋を作成する為の定義ファイル。 記述仕様については補足 1. 処方箋 CDA_2 次元シンボル処方箋変換定義 ファイルを参照。
標準用法名称変換定義マ スタファイル	標準用法コードまたは補足用法コードから標準用法名称を作成する為の定義フ ァイル。 必要に応じて各表定義 (表〇〇_*.xml) を参照する。
表〇〇_*.xml	入力ファイルの標準用法コードパターン、該当桁に応じた表を参照する。

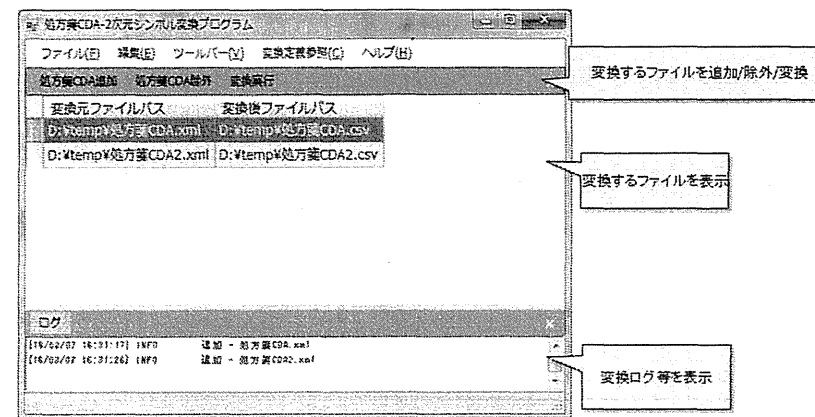
### 出力ファイル

2 次元処方箋ファイル	入力ファイルを元に作成された 2 次元シンボル処方箋 (JAHIS 院外処方せん 2 次元シンボル記録条件規約 Ver1.1)
-------------	--

## 3. 画面

プログラムの画面を以下に記述する。

### メイン画面



## 4. 機能

### 処方箋 CDA 追加

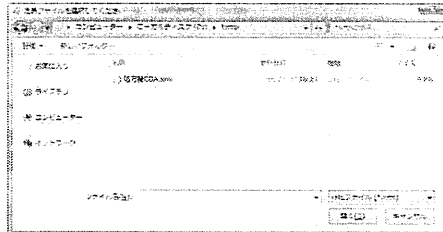
オープンファイルダイアログを表示して、変換対象の処方箋 CDA を選択する。

1. ツールバーの「処方箋 CDA 追加」をクリック

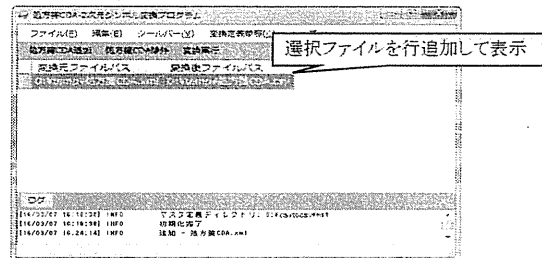
クリック

処方箋CDA追加 処方箋CDA除外 変換実行

## 2. オープンファイルダイアログで、変換対象のファイルを選択



## 3. 画面に選択ファイルを表示

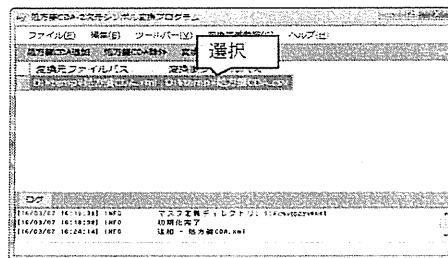


変換元ファイルパス	選択ファイルのファイルパスを表示する
変換後ファイルパス	選択ファイルの拡張子を.csvに変更したファイルパスを表示する

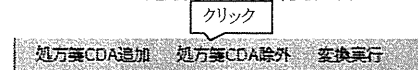
### 処方箋 CDA 除外

選択行をリストから削除する。

#### 1. 削除する行を選択



## 2. ツールバーの「処方箋 CDA 除外」をクリック



## 3. 画面から対象のファイルを削除する。(物理的なファイル削除はしない)

### 変換実行

画面に表示されている処方箋 CDA ファイルを 2 次元シンボル処方箋形式に変換する。

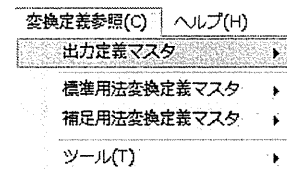
- 「処方箋 CDA 追加」で変換対象のファイルを追加
- 処方箋 CDA を追加した状態で、ツールバーの「変換実行」をクリック



- 変換処理を実行し、「変換後ファイルパス」に 2 次元シンボル処方箋が作成される。  
変換仕様については 4. 変換仕様を参照

### 変換定義・マスタ参照/保存

標準用法コードから、標準用法名称（または補足用法コードから補足用法名称）を作成する定義ファイルを参照・編集・更新する

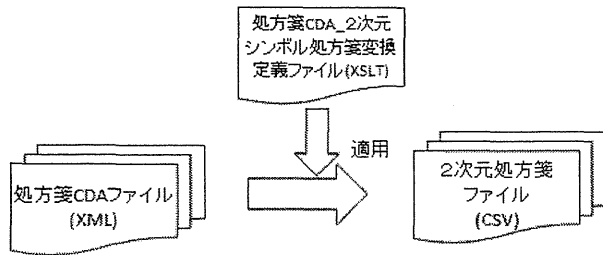


出力定義マスタ	出力定義マスタの表示を行う
標準用法変換定義マスタ	標準用法変換に使用する各表定義の表示を行う
補足用法変換定義マスタ	補足用法変換に使用する各表定義の表示を行う
ツール	変換したデータの保存を行う

## 5. 変換仕様

### 処方箋 CDA\_2次元シンボル処方箋変換仕様

処方箋 CDA の XML ファイルに XSLT スタイルシートを適用することで、2次元シンボル処方箋を作成する。



但し、この時点では下記の内容が不足または誤っているため、プログラムで適切な2次元シンボル処方箋形式に補足変換する。

- 負担区分レコードが作成できていない。
- 用法レコードの用法名称に処方箋 CDA に記載されている用法名称が記録されている
- 補足用法コードが、用法レコードとして作成される。  
※ 本来であれば用法補足レコードに作成するべきだが、コードを記録する欄が無いため。
- 不均等指示の薬品の場合、服用時点毎に不均等レコードが作成される。
- RP 毎の連番が 1 として出力される。
- 医師レコードの医師カナ氏名、患者氏名レコードの患者カナ氏名が全角で出力される。

※ 2、3 に関しては次項「標準用法名称/補足用法名称作成仕様」を参照

※ 処方箋 CDA\_2次元シンボル処方箋変換定義ファイルと、処方箋 CDA とマッピングについては[ 8. 定義ファイル]-[変換定義ファイル]参照

### 標準用法名称/補足用法名称作成仕様

#### 標準用法コードから標準用法名称の作成

指定された標準用法コードを元に、変換ルール.xml を参照し、全ての桁において、条件に当てはまる出力値を生成・取得する。

例えば、「1013044400000000」が指定された場合、1行目の「食事等タイミングを基本とする内服用法」が選択され、出力値から、「1日3回朝食後」が生成される。

#### 補足用法コードから補足用法名称の作成

指定された補足用法コードを元に、補足用法変換ルール.xml を参照し、全ての桁において、条件に当てはまる出力値を生成・取得する。

例えば、「W110000」が指定された場合、3行目の「曜日指定」が選択され、「日曜と月曜に服用」が生成される。

## 6. 定義ファイル

変換定義ファイル (xslt)	処方箋 CDA に XSLT を適用することで、2次元シンボル処方箋を作成する。 ※ 変換できない項目についてはプログラムで適切な形に修正する。
標準用法名称作成用定義ファイル (xml)	標準用法コードまたは補足用法コードから標準用法名称を作成する為の定義。

### 変換定義ファイル

XSLT によるスタイルシートを定義する。

#### XML ファイル定義

ファイル名	default.xsl
XML 定義ファイル 文字エンコード	Shift_JIS
XML version	1.0
スタイルシート バージョン	2.0
名前空間	xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
出力データ 文字エンコーディング	Shift_JIS

#### XML ヘッダ部定義サンプル

```

<?xml version="1.0" encoding="Shift_JIS"?>
<xsl:stylesheet version="2.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform" >
<xsl:output method="text" encoding="Shift_JIS" />
</xsl:stylesheet>
  
```

#### 2次元シンボル処方箋への対応付け template 定義

XPath	値
/xsl:stylesheet/xsl:template	
@match	"/"

バージョンレコード変換定義

バージョンレコードは固定値を設定する。

XPath	値
text()	JAHIS2
xsl:text/	
text()	[CRLF]

※文中の[CRLF]は改行コードを指す。

記述サンプル

```
JAHIS2<xsl:text>
</xsl:text>
```

1 医療機関レコード

・中間定義

```
記述パス
/xsl:stylesheet/xsl:template/
```

XPath	値
xsl:apply-templates	
@select	EPD/ Prescription/ ClinicalDocument/ author/ assignedAuthor/ representedOrganization/ asOrganizationPartOf/ wholeOrganization
@mode	csv1

・変換定義

```
記述パス
/xsl:stylesheet/
処方箋 CDA 参照 XPath
/EPD/Prescription/ClinicalDocument/author/assignedAuthor/representedOrganization/asOrganizationPartOf/wholeOrganization
```

XPath	値	項目名
xsl:template/		
@match	wholeOrganization	
@mode	csv1	
text()	1,	レコード番号
xsl:value-of/		
@select	substring(id/@extension,3,1)	医療機関コード種別(医科:1) 医療機関番号の3桁目を取得する
text()	,	区切り
xsl:value-of/		
@select	substring(id/@extension,4)	レセプト提出用コード 処方箋 CDA では都道府県コード(2桁)+ 医療機関コード種別(1桁)+医療機関コ ード(7桁)の10桁構成となっているため 頭から3文字削除する
text()	,	
xsl:value-of		
@select	substring(id/@extension,1,2)	都道府県番号 医療機関番号の頭2桁を取得する
xsl:value-of		
@select	name	医療機関名称
xsl:text/		
text()	[CRLF]	改行コード

```
XML記述
<xsl:template>
<xsl:apply-templates
  select="EPD/Prescription/ClinicalDocument/author/assignedAuthor/representedOrganization/asOrganizationPartOf/w
  holeOrganization"
  mode="csv1"
/>
</xsl:template>

<xsl:template
  match="wholeOrganization"
  mode="csv1">1,<xsl:value-of
  select="substring(id/@extension,3,1)" />,<xsl:value-of
  select="substring(id/@extension,4)" />,<xsl:value-of
  select="substring(id/@extension,1,2)" />,<xsl:value-of select="name" /><xsl:text>
</xsl:text></xsl:template>
```

2 医療機関所在地レコード

・中間定義

記述パス	
/xsl:stylesheet/xsl:template/	

XPath	値
xsl:apply-templates	
@select	EPD/ Prescription/ ClinicalDocument/ author/ assignedAuthor/ representedOrganization/ asOrganizationPartOf/ wholeOrganization
@mode	csv2

・変換定義

記述パス	
/xsl:stylesheet/	
処方箋 CDA 参照 XPath	
/EPD/Prescription/ClinicalDocument/author/assignedAuthor/representedOrganization/asOrganizationPartOf/wholeOrganization	

XPath	値	項目名
xsl:template/		
@match	wholeOrganization	
@mode	csv2	
text()	2,	レコード番号
xsl:value-of/		
@select	addr/postalCode	郵便番号
text()	,	
xsl:value-of/		
@select	addr/streetAddressLine	医療機関住所
xsl:text/		
text()	[CRLF]	改行コード

XML 記述	
<xsl:template>	
<xsl:apply-templates	

```

select="EPD/Prescription/ClinicalDocument/author/assignedAuthor/representedOrganization/asOrganizationPartOf/wholeOrganization"
mode="csv2"
/>
</xsl:template>

<xsl:template
match="wholeOrganization" mode="csv2">2,<xsl:value-of
select="addr/postalCode" />,<xsl:value-of
select="addr/streetAddressLine" /> <xsl:text>
</xsl:text> </xsl:template>
    
```

3 医療機関電話レコード

医療機関電話レコードは、処方箋 CDA に医療機関電話番号が記載されている場合のみ出力する。

・中間定義

記述パス	
/xsl:stylesheet/xsl:template/	

XPath	値
xsl:apply-templates	
@select	EPD/ Prescription/ ClinicalDocument/ author/ assignedAuthor/ representedOrganization/ asOrganizationPartOf/ wholeOrganization
@mode	csv3

・変換定義

記述パス	
/xsl:stylesheet/	
処方箋 CDA 参照 XPath	
/EPD/Prescription/ClinicalDocument/author/assignedAuthor/representedOrganization/asOrganizationPartOf/wholeOrganization	

XPath	値	項目名
xsl:template/		
@match	wholeOrganization	

@mode	csv3	
text()	3,	レコード番号
xsl:value-of/		
@select	substring(telecom/@value,5)	電話番号 処方箋 CDA では頭に"tel:"が含まれる為 5文字目以降を取得する。
xsl:text/		
text()	[CRLF]	改行コード

```

XML記述
<xsl:template>
<xsl:apply-templates
  select="EPD/Prescription/ClinicalDocument/author/assignedAuthor/representedOrganization/asOrganizationPartOf/w
  holeOrganization" mode="csv3"
/>
</xsl:template>

<xsl:template
  match="wholeOrganization"
  mode="csv3"><xsl:if test="telecom/@value">3,<xsl:value-of
  select="substring(telecom/@value,5)" /><xsl:text>
</xsl:text></xsl:if></xsl:template>

```

#### 4 診療科レコード

・中間定義

```

記述パス
/xsl:stylesheet/xsl:template/

```

XPath	値
xsl:apply-templates	
@select	EPD/ Prescription/ ClinicalDocument/ author/ assignedAuthor/ representedOrganization
@mode	CSV4

・変換定義

```

記述パス
/xsl:stylesheet/
処方箋 CDA 参照 XPath

```

#### /EPD/Prescription/ClinicalDocument/author/assignedAuthor/representedOrganization

XPath	値	項目名
xsl:template/		
@match	representedOrganization	
@mode	csv4	
text()	4,	レコード番号
xsl:choose/		
xsl:when/		
@test	id/@extension	処方箋 CDA に診療科コードが記載されて いる場合が記載されている場合
text()	2,	診療科コード種別 2 をセットする。
xsl:value-of		
@select	id/@extension"	診療科コード
text()	,	
xsl:value-of		
@select	name[@use='IDE']	診療科名
xsl:otherwise/		
text()	1,,	処方箋 CDA に診療科コードが記載されて いない場合は2 桁目は空欄にする。
xsl:value-of/		
@select	name[@use='IDE']	診療科名
xsl:text/		
text()	[CRLF]	改行コード

```

XML記述
<xsl:template>
<xsl:apply-templates
  select="EPD/Prescription/ClinicalDocument/author/assignedAuthor/representedOrganization " mode="csv4"
/>
</xsl:template>

<xsl:template
  match="representedOrganization"
  mode="csv4">4,<xsl:choose>
  <xsl:when test="id/@extension">2,<xsl:value-of
  select="id/@extension"/>,<xsl:value-of
  select="name[@use='IDE']"/></xsl:when>
  <xsl:otherwise>1,,<xsl:value-of
  select="name[@use='IDE']"/></xsl:otherwise>
</xsl:choose><xsl:text>
</xsl:text></xsl:template>

```



5 医師レコード

・中間定義

記述パス	
/xsl:stylesheet/xsl:template/	

XPath	値
xsl:apply-templates	
@select	EPD/ Prescription/ ClinicalDocument/ author/ assignedAuthor/ assignedPerson
@mode	csv5

・変換定義

記述パス	
/xsl:stylesheet/	
処方箋 CDA 参照 XPath	
/EPD/Prescription/ClinicalDocument/author/assignedAuthor/assignedPerson	

XPath	値	項目名
xsl:template/		
@match	assignedPerson	
@mode	csv5	
text()	5,,	レコード番号, 医師コード 医師コードは処方箋 CDA の仕様上 OID に 医療機関が含まれ可変となるため Xpath で 指定できない為、空電とする
xsl:value-of/		
@select	name[@use='SYL']	処方医師名 (漢字)
text()	,	区切り
xsl:value-of/		
@select	name[@use='IDE']	処方医師名 (カナ)
xsl:text/		
text()	[CRLF]	改行コード

処方医師名 (フリガナ) については処方箋 CDA では全角であるため、変換後にプログラムで半角に変換す

る。

XML 記述	
<pre>&lt;xsl:template&gt; &lt;xsl:apply-templates select="EPD/Prescription/ClinicalDocument/author/assignedAuthor/assignedPerson" mode="csv5" /&gt; &lt;xsl:apply-templates select="EPD/Prescription/ClinicalDocument/author/assignedAuthor/representedOrganization" mode="csv4" /&gt; &lt;/xsl:template&gt;  &lt;xsl:template match="assignedPerson" mode="csv5"&gt; 5,, &lt;xsl:value-of select="name[@use='SYL']" /&gt; , &lt;xsl:value-of select="name[@use='IDE']" /&gt; &lt;xsl:text&gt; &lt;/xsl:text&gt; &lt;/xsl:template&gt;</pre>	

11 患者氏名レコード

・中間定義

記述パス	
/xsl:stylesheet/xsl:template/	

XPath	値
xsl:apply-templates	
@select	EPD/ Prescription/ ClinicalDocument/ recordTarget/ patientRole
@mode	csv11

・変換定義

記述パス	
/xsl:stylesheet/	
処方箋 CDA 参照 XPath	
/EPD/Prescription/ClinicalDocument/recordTarget /patientRole	

XPath	値	項目名
xsl:template/		
@match	patientRole	

@mode	csv11	
text()	11,	レコード番号,
xsl:value-of/		
@select	id[substring(@root,1,21)='1.2.392.200250.3.3.1']/@extension	患者コード
text()	,	
xsl:value-of/		
@select	patient/name[@use='IDE']	患者名 (漢字)
text()	,	
xsl:value-of/		
@select	patient/name[@use='SYL']	患者名 (カナ)
xsl:text/		
text()	[CRLF]	改行コード

処方医師名 (フリガナ) については処方箋 CDA では全角であるため、変換後にプログラムで半角に変換する。

XML 記述
<pre>&lt;xsl:template&gt; &lt;xsl:apply-templates select="EPD/Prescription/ClinicalDocument/recordTarget/patientRole" mode="csv11" /&gt; &lt;/xsl:template&gt;  &lt;xsl:template   match="patientRole"   mode="csv11"&gt;11,&lt;xsl:value-of   select="id[substring(@root,1,21)='1.2.392.200250.3.3.1']/@extension"/&gt;,&lt;xsl:value-of   select="patient/name[@use='IDE']" /&gt;,&lt;xsl:value-of   select="patient/name[@use='SYL']" /&gt;&lt;xsl:text&gt; &lt;/xsl:text&gt;&lt;/xsl:template&gt;</pre>

## 12 患者性別レコード

・中間定義

記述パス
/xsl:stylesheet/xsl:template/

XPath	値
xsl:apply-templates	
@select	EPD/ Prescription/ ClinicalDocument/ recordTarget/

	patientRole
@mode	csv12

・変換定義

記述パス
/xsl:stylesheet/
処方箋 CDA 参照 XPath
/EPD/Prescription/ClinicalDocument/recordTarget /patientRole

XPath	値	項目名
xsl:template/		
@match	patientRole	
@mode	csv12	
text()	12,	レコード番号,
xsl:value-of/		
@select	translate( patient/administrativeGenderCode/@code, 'MF', '12' )	患者性別 "M"を"1"に "F"を"2"に変換する
xsl:text/		
text()	[CRLF]	改行コード

XML 記述
<pre>&lt;xsl:template&gt; &lt;xsl:apply-templates select="EPD/Prescription/ClinicalDocument/recordTarget/patientRole" mode="csv12" /&gt; &lt;/xsl:template&gt;  &lt;xsl:template   match="patientRole"   mode="csv12"&gt;12,&lt;xsl:value-of   select="translate(patient/administrativeGenderCode/@code,'MF','12')" /&gt;&lt;xsl:text&gt; &lt;/xsl:text&gt;&lt;/xsl:template&gt;</pre>

## 13 患者生年月日レコード

・中間定義

記述パス
/xsl:stylesheet/xsl:template/

XPath	値

xsl:apply-templates	
@select	EPD/ Prescription/ ClinicalDocument/ recordTarget/ patientRole
@mode	csv13

・変換定義

記述パス
/xsl:stylesheet/
処方箋 CDA 参照 XPath
/EPD/Prescription/ClinicalDocument/recordTarget/patientRole

XPath	値	項目名
xsl:template/		
@match	patientRole	
@mode	csv13	
text()	13,	レコード番号,
xsl:value-of/		
@select	patient/birthTime/@value	生年月日
xsl:text/		
text()	[CRLF]	改行コード

XML 記述
<pre>&lt;xsl:template&gt; &lt;xsl:apply-templates select="EPD/Prescription/ClinicalDocument/recordTarget/patientRole" mode="csv13" /&gt; &lt;/xsl:template&gt;  &lt;xsl:template match="patientRole" mode="csv13"&gt;13,&lt;xsl:value-of select="patient/birthTime/@value " /&gt;&lt;xsl:text&gt; &lt;/xsl:text&gt;&lt;/xsl:template&gt;</pre>

14 患者一部負担レコード

患者一部負担レコードは処方日時点の患者年齢と負担割合から求める必要があるため、プログラムで処理する。

21 保健種別レコード

・中間定義

記述パス
/xsl:stylesheet/xsl:template/

XPath	値
xsl:apply-templates	
@select	EPD/ Prescription/ ClinicalDocument/ component/ structuredBody/ component/ section/ entry[../code/@code='11']/ act/ entryRelationship[@typeCode='COMP']
@mode	csv21

・変換定義

記述パス
/xsl:stylesheet/
処方箋 CDA 参照 XPath
/EPD/Prescription/ClinicalDocument/component/structuredBody/component/section/entry[../code/@code='11']/act/entryRelationship[@typeCode='COMP']

XPath	値	項目名
xsl:template/		
@match	act	
@mode	csv21	
xsl:choose/		
xsl:when/		
@test	"code/@nullFlavor"	
text()	21,9	レコード番号 コードが未記載の場合
xsl:when/		

@test	performer/ assignedEntity/ id[@root='1.2.392.200119.6.101'] and code[@codeSystem='1.2.392.200250.2.2.13']	保険者番号を確認
text()	21,	レコード番号
xsl:value-of/		
@select	code/@code	保険種別
xsl:text/		
text()	[CRLF]	

```

XML 記述
<xsl:template>
<xsl:apply-templates
  select="EPD/Prescription/ClinicalDocument/component/structuredBody/component/section/entry[../code/@code='11']/act/entryRelationship[@typeCode='COMP']" mode="csv21" />
</xsl:template>

<xsl:template match="act" mode="csv21">
<xsl:choose>
<xsl:when
  test="code/@nullFlavor">21,9</xsl:when>
<xsl:when
  test="performer/assignedEntity/id[@root='1.2.392.200119.6.101']
  and
  code[@codeSystem='1.2.392.200250.2.2.13']>21, <xsl:value-of
  select="code/@code" /></xsl:when>
</xsl:choose> <xsl:text>
</xsl:text>
</xsl:template>

```

22 保険者番号レコード

・中間定義

記述パス	
/xsl:stylesheet/xsl:template/	

XPath	値
xsl:apply-templates	
@select	EPD/ Prescription/ ClinicalDocument//

	entry[../code/@code='11']/ act/ entryRelationship[@typeCode='COMP']/ act/ performer/ assignedEntity/ id
@mode	csv22

・変換定義

記述パス	
/xsl:stylesheet/	
処方箋 CDA 参照 XPath	
/EPD/Prescription/ClinicalDocument//entry[../code/@code='11']/act/entryRelationship	
[@typeCode='COMP']/act/performer/assignedEntity/id	

XPath	値	項目名
xsl:template/		
@match	id	
@mode	csv22	
xsl:choose/		
xsl:when/		
@test	../@root='1.2.392.200119.6.101'	
text()	22,	レコード番号
xsl:value-of/		
@select	../@extension	保険者番号
xsl:when		
@test	../@nullFlavor	
text()	22,	レコード番号
xsl:text		
text()	[CRLF]	

```

XML 記述
<xsl:template>
</xsl:text> <xsl:apply-templates
  select="EPD/Prescription/ClinicalDocument//entry[../code/@code='11']/act/entryRelationship[@typeCode='COMP']/
  act/performer/assignedEntity/id" mode="csv22" /> <xsl:text>
</xsl:template>

<xsl:template match="id" mode="csv22">

```