

日医標準レセプトソフト クラウド版  
カスタムバッチ登録手順

ベンダー向け

2019年9月30日

日本医師会 ORCA 管理機構株式会社

## 目次

1	概要	2
2	事前準備	4
2.1	ソフトウェアのインストール	4
2.2	証明書の用意とサポート事業者 ID の確認	4
2.3	実行するプログラム	4
2.4	Dockerfile の作成	4
2.5	実行可能なテナントの設定	5
2.6	制限事項	5
3	操作の流れ	8
3.1	カスタムバッチ用レポジトリの作成	8
3.2	カスタムバッチの登録とビルド	8
3.3	ローカルレポジトリの作成	9
3.4	プログラムの追加	9
3.5	ビルド結果の確認	9
4	医療機関の設定	11
4.1	日次統計	11
4.2	月次統計	12
4.3	県単独事業情報	12
4.4	入院定期請求	14
4.5	ユーザプログラム	14

# 1 概要

---

本資料は、医療機関(テナント)がカスタムバッチを実行するための準備としてベンダーが行う手順を記述する。

標準バッチは日レセクラウド上の日レセサービス上で実行される。しかし、カスタムバッチは ORCA 管理機構とは開発元が異なるプログラムのため、安全に実行するために日レセサービスとは別の日レセクラウド上の隔離された環境で実行される。

日レセクラウド上のカスタムバッチの実行には Docker(<https://www.docker.com/>)というソフトウェアを用いている。カスタムバッチは Docker コンテナとして実行される。実行のためには事前の準備として、実行するカスタムバッチのソースコードを日レセクラウド上にアップロードし、それを元に Docker イメージとして構築する必要がある。

カスタムバッチ向けの Docker イメージを作成するためには、1.3 節および 1.4 節で説明するカスタムバッチ向けのカスタムバッチ用レポジトリを用意する必要がある。

Docker イメージ 1 つに対して、1 つのレポジトリを用いる。カスタムバッチ用レポジトリは git レポジトリになっており、ベンダーが開発したソースコードをコミットする。日レセクラウド上のリモートレポジトリに push を行うと、Docker イメージの構築が開始される。ベンダーは直接ビルドを行う必要はない。

クライアントからカスタムバッチを実行する際は、レポジトリの情報とカスタムバッチ内のプログラム名やシェルスクリプトパスを元に実行を行う。個別の設定については 1.5 節を参照。Docker イメージ内に指定したプログラムやシェルスクリプトが存在すればカスタムバッチが実行できる。Docker イメージ内のプログラム数に制限はない。そのため、複数のカスタムバッチを実行できる共通利用のためのレポジトリ(Docker イメージ)を 1 つ用意し、同じレポジトリのパスを切り替えて利用することもできる。この場合は 1 つのレポジトリのみを用いればよい。しかし、同名のプログラムを医療機関ごとに切り替えて実行したい場合は複数のレポジトリを作成し、レポジトリを切り替えて利用する必要がある。1 ベンダーはカスタムバッチ用レポジトリを 100 件まで作成する事ができる。ただし、日レセクラウド上の設定のため、この上限は変更される可能性がある。

システム管理者が行う手順については「カスタムバッチ利用手順(システム管理者用)」を参照のこと。未設定の場合は県単独事業の総活用・公費請求書作成以外のカスタムバッチの実行を行う事ができない。クライアントのレポジトリ設定が完了した後

の実行手順は標準バッチと同じのため、ユーザ向けには日医標準レセプトソフトマニュアル(<https://manual.orca.med.or.jp>)を参照のこと。

## 2 事前準備

---

### 2.1 ソフトウェアのインストール

以下のソフトウェアをあらかじめインストールする。

- git

### 2.2 証明書を用意とサポート事業者 ID の確認

システム管理者から証明書と自身のサポート事業者 ID を受け取る。

以下、サポート事業者 ID が s00000001 の test\_vendor@example.com のベンダーの証明書の名前の例。XXX にはシステム管理上の数値が入る。

- 000XXX\_JP\_s00000001\_test\_vendor@example.com.crt
- 000XXX\_JP\_s00000001\_test\_vendor@example.com.enc.pem
- 000XXX\_JP\_s00000001\_test\_vendor@example.com.p12
- 000XXX\_JP\_s00000001\_test\_vendor@example.com.pass
- 000XXX\_JP\_s00000001\_test\_vendor@example.com.pem
- ca.crt

### 2.3 実行するプログラム

実行するカスタムバッチのソースコードを用意する。

以降の節では、日レセクラウド上で特別に用意するファイルと実行するプログラムの制限事項について説明する。

### 2.4 Dockerfile の作成

カスタムバッチの実行のためには Dockerfile に以下の記述を行う必要がある。

- ベースイメージの指定: FROM ginbee\_cb\_base を指定。
  - 日レセ 5.1 関連のパッケージがインストールされた Ubuntu のイメージ。
- ファイルの追加: 上述のプログラムを Docker イメージ内に ADD 命令にて追加。

- ベースイメージが日レセの環境のため、追加するファイルはオンプレ版の  
カスタムバッチ同様のパスに配置するようにする必要がある。
- タイムアウトの延長: (必要であれば) 上述の `GINBEE_CUSTOM_BATCH_TIMEOUT` 環境変数を `ENV` 命令にて追加。
- (必要であれば) パッケージのインストールや設定(例: `/etc/jma-receipt/kentan.inc`) の更新など。
- (必要であれば) `site-upgrade.sh` の実行。

以下にその例(日計表)を示す。

```
FROM ginbee_cb_base  
  
ADD cobol/A32001BD01.CBL /usr/local/site-jma-receipt/cobol/  
ADD cobol/copy/A32001HD01.INC /usr/local/site-jma-receipt/cobol/copy/  
ADD form/A32001HD01.red /usr/local/site-jma-receipt/form/  
RUN sudo -u orca /usr/lib/jma-receipt/scripts/allways/site-upgrade.sh
```

## 2.5 実行可能なテナントの設定

レポジトリの最上位ディレクトリに `authorized_tenants` という名前のファイルを用意することで、該当のカスタムバッチを実行可能なテナントをベンダー側で設定する事ができる。用意しない場合はベンダーに所属する全てのテナントが該当のカスタムバッチを実行可能となる。

`authorized_tenants` ファイルには以下のようにテナントの ID を改行区切りで指定する。

```
1  
2  
3
```

テナント ID はテナント管理画面から参照する事ができる。

## 2.6 制限事項

オンプレ版のカスタムバッチとは異なり、日レセクラウド上のカスタムバッチでは以下の処理を行う事ができない。

- カスタムバッチからの外部サイトへの通信
  - ただし、カスタムバッチを実行したテナント自身の DB のうち、以下のテーブルには書き込みが可能。括弧内の表記があるものはそのテーブルを更新するサブプログラム名。
    - tbl\_jobkanri(ORCSJOB)
    - tbl\_prtkanri(ORCSPRT)
    - tbl\_prtdata(ORCSPRT)
    - tbl\_prvkanri(ORCSPRT)
    - tbl\_toukeicsv\_h(ORCSTOUKEICSV)
    - tbl\_toukeicsv\_b(ORCSTOUKEICSV)
    - tbl\_csv\_info(ORCSCSVINFO)
    - tbl\_file\_info(ORCSFILESV)
    - tbl\_seikyu\_etc
    - tbl\_seikyu\_koh
    - tbl\_seikyu\_main
    - monblob
    - monbatch
    - monbatch\_clog
    - monbatch\_log
  - 上記以外のテーブルはテナント自身の DB であれば読み込みは可能
- カスタムバッチからの印刷
  - ただし、日次統計、月次統計処理についてはクライアントが印刷まで行うためカスタムバッチ内から対応する必要はない。クライアントの印刷は最後に行われるため、カスタムバッチの印刷物は標準バッチの印刷物の後に印刷される。

カスタムバッチは以下の必要最低限の環境変数を受け取って実行される。内容はオンプレ版と同様。このため、他の環境変数を参照している場合は問題なく動作するよう注意する必要がある。

- MCP\_TENANT
- MCP\_LOGFILE\_PREFIX
- MCP\_MIDDLEWARE\_NAME
- MCP\_TEMPDIR
- MCP\_TEMPDIR\_ROOT

- MONDB\_HOST
- MONDB\_PORT
- MONDB\_NAME
- MONDB\_USER
- MONDB\_PASS
- MONDB\_ENC\_PASS
- MON\_DIRECTORY\_PATH
- COB\_LIBRARY\_PATH

また、日レセクラウド独自に以下の環境変数を受け取る。

名前	説明
MONBATCH_PRINTABLE	バッチから印刷処理が可能かどうか(docker コンテナ内で実行されたバッチかどうか)。可能な場合は 1 か未定義のいずれか。可能でない場合は 0。実行するプログラムをオンプレ版と共有する場合、分岐に使用する事ができる。
GINBEE_CUSTOM_BATCH_TIMEOUT	カスタムバッチのタイムアウト時間。省略した場合は 180 秒。後述の Dockerfile 内に記述する事を想定。180 秒以上かかるバッチ処理に設定する。
GINBEE_CUSTOM_BATCH_OPTION_ENABLED	カスタムバッチを実行するテナントの追加オプションの契約状況。無効: 0、有効: 1 のいずれか。

## 3 操作の流れ

---

ベンダーはカスタムバッチを医療機関(テナント)で実行させるためには開発したプログラムを日レセクラウド上にコピーする必要がある。

1. カスタムバッチ用レポジトリの作成
2. カスタムバッチの登録とビルド
3. 医療機関(テナント)の設定

カスタムバッチの実行については標準バッチと同様の手順となるため、省略する。

ベンダーがカスタムバッチの開発時や障害発生時には日レセクラウド上でバッチ処理を実行した際のログはテナントログ画面から確認する事ができる。

### 3.1 カスタムバッチ用レポジトリの作成

ベンダーは、医療機関(テナント)が利用できるよう以下の操作を行う。

1. システム管理サイトにログインする。  
VPN 経由の場合・・・<https://ctrl.cmo.orcamo.jp/>  
TLS1.2 接続サービスの場合・・・<https://ctrl.glcmo.orcamo.jp/>
2. [ベンダー情報]をクリックする。
3. 連携アプリ設定の[日レセ管理画面]をクリックする。
4. 右上のメニューの「カスタムバッチ用レポジトリ」をクリックする。
5. 「新規作成」をクリックする。
6. レポジトリ名と説明の項目を入力して「登録する」をクリックする。

登録後にレポジトリ名をクリックして詳細画面にてレポジトリ URL を確認して次の手順に進む。

### 3.2 カスタムバッチの登録とビルド

1. ローカルレポジトリの作成
2. プログラムの追加
3. コミットしたプログラムの push
4. ビルド結果の確認

以降の手順はコマンドプロンプト上で実行する。

ここからは以下の前提を元に作業手順を記述する。コマンドを実際に実行する際は自身の情報に読み替えて実行する必要がある。

- 作業ディレクトリ : ~/cb/
- 証明書が置いているディレクトリ : ~/ssl/
- レポジトリ名: repos
- カスタムバッチ向けのレポジトリ URL:  
VPN 経由の場合  
https://repos.orca.orcamo.jp/custom\_batch/vendor??/repos  
TLS1.2 接続サービスの場合  
https://repos.glorca.orcamo.jp/custom\_batch/vendor??/repos

### 3.3 ローカルレポジトリの作成

sms で作成したレポジトリに対して、開発したカスタムバッチを push するためのローカルレポジトリを作成する。

```
$ cd ~/cb
$ git init
$ git remote add origin https://repos.orca.orcamo.jp/custom_batch/vendor??/repos
$ git config --local http.sslCert ~/ssl/000XXX_JP_s00000001_test_vendor@example.com.crt
$ git config --local http.sslKey ~/ssl/000XXX_JP_s00000001_test_vendor@example.com.pem
$ git config --local http.sslCaInfo ~/ssl/ca.crt
```

### 3.4 プログラムの追加

```
$ cd ~/cb
$ git pull origin master
$ git add *
$ git commit -m "comment"
$ git push -u origin master
```

### 3.5 ビルド結果の確認

前項までの作業から数分程度後で、システム管理サイトのカスタムバッチレポジトリの管理画面から、該当のレポジトリの「ビルド履歴」を確認する。この画面では、

Docker によってビルドされたイメージのビルドログやビルド成否を確認する事ができる。

```
状態: 成功
終了コード: 0
gitリビジョン: cae96596f62f1d1e7a0bcd7a5c98c6a3e8daf596
ビルド開始日時: 2017/04/20 10:59:23
ビルドログ:
Sending build context to Docker daemon 660.5 kB

Step 1/6 : FROM ginbee_cb_base
----> d39c015cea50
Step 2/6 : ADD SCSAMPLE01.CBL /usr/local/site-jma-receipt/cobol/
----> 216cc6276b1d
Removing intermediate container 29c2035e5531
Step 3/6 : ADD SCSAMPLE01.INC /usr/local/site-jma-receipt/cobol/copy/
----> f1072bc4a19b
Removing intermediate container a353a5c1c80f
Step 4/6 : ADD SCSAMPLE01.red /usr/lib/jma-receipt/form/
----> b1ad1870fb29
Removing intermediate container 71858b5b0d88
Step 5/6 : RUN sudo -u orca /usr/lib/jma-receipt/scripts/always/site-upgrade.sh
----> Running in 5b844e9b005f
Building SCSAMPLE01.so...done
Copying scripts files...done
Copying ldef files...done
Copying data files...done
Copying etc files...done
----> 330115001fb3
Removing intermediate container 5b844e9b005f
Step 6/6 : CMD /bin/bash
----> Running in 34d018b7dca0
----> 8c2b954e4c00
Removing intermediate container 34d018b7dca0
Successfully built 8c2b954e4c00
```

ビルドに成功した場合は日レセクラウド上でカスタムバッチを実行する準備が整った状態となる。

## 4 医療機関の設定

---

日レセクラウドでは、以下の医療事務において標準バッチ以外にベンダー供給のカスタムバッチを実行する事ができる。

- 日次統計
- 月次統計
- 県単独事業情報
- 入院定期請求
- ユーザプログラム

それぞれで上述のビルド手順でカスタムバッチのビルドまで完了した後はクライアントにてレポジトリの設定を行う事でカスタムバッチの実行が可能になる。カスタムバッチの実行については標準バッチと同様の手順となるため、省略する。

### 4.1 日次統計

以下の操作を行う。

1. ログインする。
2. **01 医療事務**をクリックする。
3. **91 マスタ登録**をクリックする。
4. **101 システム管理マスタ**をクリックする。
5. **システム管理情報設定画面**で以下の操作を行う。
  1. 管理コードに **3001 統計帳票出力情報（日次）** を選択してエンターキーを押下する。
  2. 標準バッチと同様に区分コード、選択番号、有効年月日を入力する。
  3. **確定**ボタンをクリックする。
6. **システム管理情報-統計帳票出力情報画面**で以下の操作を行う。
  1. 標準バッチと同様に帳票名、帳票パラメタ、パラメタ説明を入力する。
  2. プログラム名、レポジトリ名を入力する(以下は入力例)。
    - レポジトリ名: vendor1/nikkeihyo
    - プログラム名: A32001BD01
  3. **登録**ボタンをクリックする。



日次統計

有効期間 00000000 ~ 99999999

帳票番号 001  前回入力されたパラメタを初期設定する ※日付は対象外です

帳票名 サンプル帳票 (日次)

プログラム名 A32001ED01

レポジトリ名 vendor1/nikkeihyo

帳票パラメタ	1	診療日	YMD	<input checked="" type="radio"/> 必須	<input type="radio"/> 任意
2				<input checked="" type="radio"/> 必須	<input type="radio"/> 任意
3				<input checked="" type="radio"/> 必須	<input type="radio"/> 任意
4				<input checked="" type="radio"/> 必須	<input type="radio"/> 任意
5				<input checked="" type="radio"/> 必須	<input type="radio"/> 任意
6				<input checked="" type="radio"/> 必須	<input type="radio"/> 任意
7				<input checked="" type="radio"/> 必須	<input type="radio"/> 任意
8				<input checked="" type="radio"/> 必須	<input type="radio"/> 任意

パラメタ説明 診療日 診療日を入力します。

戻る クリア 削除 書き 登録

## 4.2 月次統計

1. ログインする。
2. **01 医療事務**をクリックする。
3. **91 マスタ登録**をクリックする。
4. **101 システム管理マスタ**をクリックする。
5. システム管理情報設定画面で以下の操作を行う。
  1. 管理コードに **3002 統計帳票出力情報 (月次)** を選択してエンターキーを押下する。
  2. 標準バッチと同様に区分コード、選択番号、有効年月日を入力する。
  3. **確定**ボタンをクリックする。
6. システム管理情報-統計帳票出力情報画面で日次統計と同様の操作を行う。

## 4.3 県単独事業情報

1. ログインする。
2. **01 医療事務**をクリックする。
3. **91 マスタ登録**をクリックする。

4. **101 システム管理マスタ**をクリックする。
5. システム管理情報設定画面で以下の操作を行う。
  1. 管理コードに **3004 県単独事業情報**を選択してエンターキーを押下する。
  2. 標準バッチと同様に有効年月日を入力する。
  3. **確定**ボタンをクリックする。
6. システム管理情報-県単独情報設定画面で以下の操作を行う。
  1. 標準バッチと同様にシェル名、処理名称、処理対象、プリンタ名、パラメタ種別を入力し、**保存**する(以下は入力例)。
    - シェル名: `seikyu3205.sh`
  2. 一番右側のボタンをクリックしてレポジトリ名を入力する(以下は入力例)。
    - `vendor1/kentan_shimane`
  3. **確定**ボタンをクリックする。

コード	シェル名	処理名称	処理対象				プリンタ名	パラメタ種別
			1	2	3	4		
0001	sokatu3202.sh	決定通知書(国保)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
0002	sokatu3203.sh	国保診療報酬請求書	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
0003	seikyu3205.sh	乳児医療費請求書	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	1 保険者番号	
0004	seikyu3206.sh	福祉医療費請求書	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	1 保険者番号	
0005			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
0006	seikyu3209b2.sh	精神医療費請求書(雲南町)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
0007	seikyu3210b2.sh	精神医療費請求書(出雲市)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
0008	seikyu3211b2.sh		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
0009	seikyu3212b2.sh		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
0010	seikyu3217b2.sh	レポジトリ名: vendor1/kentan_shimane	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
0011	seikyu3218.sh		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
0012			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
0013			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
0014			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
0015			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
0016			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
0017	sokatu3204.sh	高根県診療報酬請求書(広域連合)	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
0018			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
0019			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
0020			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
0021			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
0022			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
0023			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
0024			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
0025			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		

7. **44 総活用・公費請求書作成画面**の実行に必要な設定を追加で実施する。
  - 標準バッチと同様の設定のため、設定の詳細は省略

- 例えば該当のカスタムバッチが乳児医療請求書に関するものであれば、**104 保険番号マスタ**から乳児医療費に関する設定等を行う必要がある。

## 4.4 入院定期請求

1. ログインする。
2. **01 医療事務**をクリックする。
3. **91 マスタ登録**をクリックする。
4. **101 システム管理マスタ**をクリックする。
5. システム管理情報設定画面で以下の操作を行う。
  1. 管理コードに **5010 定期請求情報**を選択してエンターキーを押下する。
  2. **確定**ボタンをクリックする。
6. 定期請求登録指示画面で以下の操作を行う。
  1. 標準バッチと同様に請求日から出力順までの項目を入力する。
  2. プログラム名、レポジトリ名を入力する(以下は入力例)。
    - レポジトリ名: vendor1/nyuin-teiki
    - プログラム名(請求書兼領収書): A32001HN03V02
    - プログラム名(診療費明細書): A32100HN04
  3. **確定**ボタンをクリックする。

## 4.5 ユーザプログラム

1. ログインする。
2. **01 医療事務**をクリックする。
3. **91 マスタ登録**をクリックする。
4. **101 システム管理マスタ**をクリックする。
5. システム管理情報設定画面で以下の操作を行う。
  1. 管理コードに **9700 ユーザプログラム起動情報**を選択してエンターキーを押下する。
  2. **確定**ボタンをクリックする。
6. システム管理情報-ユーザプログラム起動設定画面で以下の操作を行う。
  1. 標準バッチと同様に処理・帳票名、実行スクリプト、プリンタ名を入力する。
  2. レポジトリ名を入力する(以下は入力例)。ul style="list-style-type: none;">  - レポジトリ名: vendor1/up

3. **登録** ボタンをクリックする。