

共通クライアント
利用解説書
Linux 版

Ver 1.0

2020 年 4 月 21 日

(一社) 電子情報技術産業協会 (JEITA) EC センター

Copyright 2020 Japan Electronics and Information Technology Industries Association, EC Center All Rights Reserved

変更履歴表

項番	版数	発行／改訂日	共通クライアント 対応 Ver	変更理由	変更箇所	備考
1	1.0	2020/04/21	Ver2.3 対応	Linux 版新規作成		

目次

1	はじめに	1
2	動作環境	2
3	インストール	3
3.1	Java 実行環境のインストール	3
3.1.1	OpenJDK インストール	3
3.2	共通クライアントのインストール	4
3.2.1	インストール	4
3.2.2	ファイル/フォルダ構成	4
4	動作設定	6
4.1	設定ファイル (initfile.xml)	6
4.1.1	設定ファイルの記述内容	6
4.1.2	ebMSv3 通信機能定義ファイル (ebmsv3.properties)	8
4.1.3	JEITA 共通クライアントにおける固定パラメータ	10
4.2	利用可能ファイル名	10
4.2.1	送信ファイル	10
4.2.2	受信ファイル	11
4.3	ASP から証明書のインストールを依頼された場合	11
4.3.1	証明書の取得	11
4.3.2	証明書のインストール	12
5	起動方法	13
5.1	通信実行	13
5.1.1	シェルスクリプトによる実行	13
5.1.2	Java API による実行	14
5.2	戻り値とエラー処理方針	15
5.3	送受信処理	16
5.3.1	送信処理の動作	16
5.3.2	受信処理の動作	17
5.3.3	制限事項	17
5.4	ログファイル	18
5.4.1	ログファイルの種類	18
5.4.2	トレースログ	18
5.4.3	エラーログ	18
5.4.4	送受信結果一覧ログ	18
5.5	複数接続先と通信を行う場合	19
5.5.1	設定	19
5.5.2	送受信	19
6	トラブルシューティング	20

6.1	実行環境.....	20
6.2	送信／受信共通	20
6.3	送信処理.....	22
6.4	受信処理.....	22
7	ログ／エラーメッセージ一覧	23
7.1	ERROR レベルのログに出力されるメッセージ.....	23
7.2	Info レベルのログに出力されるメッセージ.....	24
7.3	Debug レベルのログに出力されるメッセージ	25

1 はじめに

本書では、共通クライアントのインストール、設定、機能等について記述している。

2 動作環境

OpenJDK、Oracle Java などの Java 実行環境が必要である。確認済みの共通クライアント検証環境について以下に示す。

項目	環境	
対応 OS / Java	Linux 7.0	OpenJDK 11.0.2
ディスク容量	20M バイト（通信するデータ容量によって異なる）	

表 2.1 検証環境

3 インストール

3.1 Java 実行環境のインストール

OpenJDK のインストールが必要である。

3.1.1 OpenJDK インストール

- (1) OpenJDK のサイトから java インストーラをダウンロードする。
- (2) ダウンロードしたファイルをインストールするマシンにコピーする。
- (3) インストールディレクトリを決定し、ファイルを展開する。

(実行コマンド例)

```
mkdir /usr/OpenJDK
cd /usr/OpenJDK
tar zxvf openjdk-11.0.2_linux-x64_bin.tar.gz
```

- (4) Java の環境変数を設定する。

コンソールから以下を実行する。

```
export JAVA_HOME=(java インストール先ディレクトリ)
export PATH=$JAVA_HOME/bin:$PATH
```

(実行コマンド例)

```
export JAVA_HOME=/usr/OpenJDK/jdk-11.0.2
export PATH=$JAVA_HOME/bin:$PATH
```

以上で java のインストールは完了である。

3.2 共通クライアントのインストール

3.2.1 インストール

- (1) 共通クライアントを/opt 配下にインストールする。

共通クライアントを実行するユーザでログインし、インストール先に移動する。

```
cd /opt
```

- (2) 共通クライアントインストールモジュール jeita_ediclient.tar.gz を、/opt 配下にコピーする。

- (3) 共通クライアントをインストールする。

```
tar zxvf jeita_ediclient.tar.gz
```

/opt 配下に jeita ディレクトリが作成され、コピーが始まる。

コピーが終わると、コンソールが入力状態になる。

以上で共通クライアントのインストールは完了である。

3.2.2 ファイル/フォルダ構成

解凍後のファイル/フォルダ構成は以下の通りである。

```
/opt/jeita/ediclient
├─bin
├─conf
├─data
├─lib
├─license
└─sample
```

作成されるフォルダおよびファイルについて以下に説明する。

data フォルダ、**sample** フォルダは、送受信時に使用するフォルダおよび、設定ファイル（「4. 動作設定」参照）のサンプルである。

また、**tmp** フォルダは解凍直後は作成されない。実際の動作時に自動作成される。

フォルダ名・ファイル名	説明
/opt/jeita/ediclient	EDIクライアントのインストールディレクトリ
bin	
ediclient.sh	EDIクライアントの実行ファイル 業務APはこのshファイルをコマンド実行する
conf	定義ファイルを格納するディレクトリ
ebmsv3.properties	ebMSv3通信機能定義ファイル
jeita-log4j.xml	log4jの制御用定義ファイル
data	
receive	受信ディレクトリのサンプル
send	送信ディレクトリのサンプル
send_complete	送信完了ディレクトリのサンプル
send_fail	送信失敗ディレクトリのサンプル
lib	ライブラリを格納するディレクトリ
activation.jar	JavaBeans Activation Framework
axis-ant.jar	Apache Axisライブラリ
axis-schema.jar	Apache Axisライブラリが使用するモジュール
axis.jar	Apache Axisライブラリが使用するモジュール
axis.jar.bak	Apache Axisライブラリが使用するモジュール
commons-discovery-0.2.jar	Apache Axisライブラリが使用するモジュール
commons-lang-2.1.jar	Apache Axisライブラリが使用するモジュール
commons-logging-1.0.4.jar	Apache Axisライブラリが使用するモジュール
jaxrpc.jar	Apache Axisライブラリが使用するモジュール
jeita-ebmsv3.jar	ebMSv3通信モジュール
jeita-ediclient.jar	EDIクライアントモジュール
jeita-ediclient-adaptor-ebmsv3.jar	ediclient-ebmsv3アダプタモジュール
log4j-1.2.12.jar	Log4j
mail.jar	JavaMail
saaj.jar	Apache Axisライブラリが使用するモジュール
wsdl4j-1.5.1.jar	Apache Axisライブラリが使用するモジュール
xalan.jar	Xpath/XSLTプロセッサ
xercesImpl.jar	XMLパーサ
xml-apis.jar	XMLパーサ
license	EDIクライアントが使用しているソフトウェアのライセンスファイルを格納するディレクトリ
axis.LICENSE.txt	axisライセンスファイル
log4j.LICENSE.txt	log4jライセンスファイル
xalan.LICENSE.txt	xalanライセンスファイル
xercesImpl.LICENSE.txt	xercesライセンスファイル
sample	
initfile.xml	設定ファイルのサンプル
tmp	EDIクライアント実行時の一時ファイルを出力するディレクトリ ※一時ファイルは、次回起動時に、1日経過したものは削除されますので、ユーザは特に意識する必要はありません。

表 3.1 ファイル/ディレクトリ構成

4 動作設定

4.1 設定ファイル (initfile.xml)

共通クライアントの動作に必要な共通的な情報 (ID/パスワードなど) をあらかじめ記述しておくためのファイルである。

設定ファイルのサンプルは、sample フォルダ (「3.2.2 ファイル/フォルダ構成」参照) に作成されている。

設定ファイルは、共通クライアント実行時に引数[-initfile]で指定する必要がある。(「5 起動方法」参照)

設定ファイルを複数用意し、送受信コマンド実行時に指定することにより複数の接続先との通信が可能である。ただし、送受信ディレクトリについて混在して使用しないように注意が必要である。

4.1.1 設定ファイルの記述内容

設定ファイルの形式、例は以下のとおりである。設定ファイルに記述する情報のうち、送信/受信フォルダ・ボックス ID (「表 4.1 設定ファイルの記述内容」の必須欄が○の項目) は実行時に与える引数によって上書きすることが可能である。

詳細は、「5.1 通信実行」を参照。

◎：設定ファイル中に必ず記述しなければならない項目。

○：起動引数、もしくは設定ファイル中に記述しなければならない項目。

●：オプション項目だが、動作環境によっては必ず設定しなければならない項目。

1階層タグ	2階層タグ	必須	意味	設定説明・推奨値	例
アカウント情報 account	User	◎	サーバにログインする アカウント	ASP プロバイダーから 通知された値を設定 する。	UserId
	password	◎	サーバにログインする パスワード(平文)		UserPassword
	url	◎	サーバの url	ASP プロバイダーから 通知された url を設定 する。	https://www.jeita. jp/JEITA/services /EbMSv3Server
プロキシ情報 (オプション) proxyinfo	proxy_host	●	プロキシサーバホスト名 (省略時はプロキシは 使わない)	インターネット接続に プロキシサーバがある 場合は、その情報を 設定する。プロキシサ ーバを持たない場合	ProxyHost
	proxy_port	●	プロキシポート番号(省 略時は 8080)		8080

	proxy_user	●	プロキシユーザ (省略時は認証なし)	は設定しなくていい。(空タグ)	ProxyUser	
	proxy_password	●	プロキシパスワード(平文)		ProxyPassword	
ファイル情報 fileinfo	send_directory	○	送信フォルダパス	使用するフォルダを指定する。	/opt/jeita/edicien t/data/send	
	sent_directory		送信完了フォルダパス。省略時は、送信フォルダの親フォルダ+“/sent”という名前のフォルダとなる。		/opt/jeita/edicien t/data/send_comp lete	
	senderr_directory		送信失敗フォルダ 省略時は、送信フォルダの親フォルダ+“/senderr”という名前のフォルダとなる。		/opt/jeita/edicien t/data/send_fail	
	receive_directory	○	受信フォルダパス		/opt/jeita/edicien t/data/receive	
	receive_max_messages		受信最大メッセージ数 省略時は“1”となる。 “0”を指定した場合は、すべて受信するまで受信処理を続ける。		1回の受信処理で受信されるメッセージの最大数。	1
	sendboxid	○	送信ボックス ID		送信処理時の対象ボックス ID。引数によって上書き可能。	Document
	receiveboxid	○	受信ボックスID		受信処理時の対象ボックス ID。引数によって上書き可能。	Document
ログ情報 log	error_log	◎	エラーログを保存するファイルパス	ログを出力したい適切な場所を決定し、そのファイルパスを指定する。	/opt/jeita/error.log	
	trace_log	◎	トレースログを保存するファイルパス		/opt/jeita/trace.log	

	level		トレースログに出力されるレベル。 ・Error ・Info(デフォルト) ・Debug	通常運用時は、 "Info"を設定することを推奨する。	Info(通常モード)
プロキシサーバへのCONNECT電文改行コード proxyconnect_lineseparator	—		プロキシサーバへのCONNECT電文の改行コード ・CRLF(デフォルト) ・LF (上記以外の値指定、及び空タグ指定不可)	CONNECT 電文送信時の改行コードを指定する場合に指定する。	CRLF

表 4.1 設定ファイルの記述内容

```

<?xml version="1.0" encoding="UTF-8" ?>
<setting>
  <account>
    <user>UserId</user>
    <password>UserPassword</password>
    <url>https://www.jeita.jp/JEITA/services/EbMSv3Server</url>
  </account>
  <proxyinfo>
    <proxy_host>ProxyHost</proxy_host>
    <proxy_port>8080</proxy_port>
    <proxy_user>ProxyUser</proxy_user>
    <proxy_password>ProxyPassword</proxy_password>
  </proxyinfo>
  <fileinfo>
    <send_directory>/opt/jeita/ediclient/data/send</send_directory>
    <sent_directory>/opt/jeita/ediclient/data/send_complete</sent_directory>
    <senderr_directory>/opt/jeita/ediclient/data/send_fail</senderr_directory>
    <receive_directory>/opt/jeita/ediclient/data/receive</receive_directory>
    <receive_max_messages>1</receive_max_messages>
    <sendboxid>Document</sendboxid>
    <receiveboxid>Document</receiveboxid>
  </fileinfo>
  <log>
    <error_log>/opt/jeita/error.log</error_log>
    <trace_log>/opt/jeita/trace.log</trace_log>
    <level>Info</level>
  </log>
  <proxyconnect_lineseparator>CRLF</proxyconnect_lineseparator>
</setting>

```

図 4.1 設定ファイル例

4.1.2 ebMSv3 通信機能定義ファイル (ebmsv3.properties))

送受信機能で使用する設定値を定義するファイルである。形式は Java のプロパティファイルとし、その内容は下記である。

キー	意味	記述例
java.net.ssl.trustStore	SSL 認証で使用する CA 証明書がインポートされたトラストストアファイルのパス	java.net.ssl.trustStore=/usr/OpenJDK/jdk-11.0.2/lib/security/cacerts
java.net.ssl.trustStorePassword	上記トラストストアファイルのパスワード	java.net.ssl.trustStorePassword=changeit
ebmsv3.cpadir	CPA ファイルが格納されているフォルダをしている (CPA ファイルは jeita - ebmsv3.jar に格納されている)	ebmsv3.cpadir=/opt/jeita/edicient/lib
ebmsv3.connection.timeout	強制的に通信タイムアウトする時間(秒) 指定されていない場合のデフォルトは 600 秒。また、600 秒未満の値を指定しても 600 秒となる。	ebmsv3.connection.timeout=1000
retry.count	通信をリトライする回数(回数には最初の通信も含む) 例：3 を設定した場合は、最初の通信・リトライ 2 回を行う 1~10 の整数が指定可能 1~10 以外の値を設定した場合、または指定がない場合のデフォルトは 3 回	retry.count=3
retry.interval	通信をリトライする間隔(秒) 0~60 の整数が指定可能 0~60 以外の値を設定した場合、または指定がない場合 デフォルトは 1 秒	retry.interval=1
edicient.size.max.send.file	送信ファイルの送信可能な最大サイズ(M) 指定されていない場合のデフォルトは 100M バイト	edicient.size.max.send.file=100
edicient.size.max.all.file	送信ファイルの添付ファイルも含めた全体の送信可能な最大サイズ(M) 指定されていない場合のデフォルトは 200M バイト	edicient.size.max.all.file=200

表 4.2 ebMSv3 通信機能定義ファイルの記述内容

```
## Copyright (C) 2006 Japan Electronics and Information Technology Industries Association, EC
Center All Rights Reserved

#
# JSSE TrustManager keystore
#
javax.net.ssl.trustStore=/usr/OpenJDK/jdk-11.0.2/lib/security/cacerts
javax.net.ssl.trustStorePassword=changeit

#
# cpa dir path
#
ebmsv3.cpadir=/opt/jeita/edIClient/lib
```

図 4.2 ebMSv3 通信機能定義ファイル例

4.1.3 JEITA 共通クライアントにおける固定パラメータ

ebMS 通信パラメータである ebMS Action の仕様は下記の通りである。一般の ebMS サーバとの通信において、ebMS サーバ側で調整を行うように依頼する必要がある。

JEITA 共通クライアント送信時 : PutMessage

※JEITA 共通クライアント受信時はチェックしないため任意。 GetMessage 推奨

4.2 利用可能ファイル名

送受信ファイル名、及び添付ファイル名に利用できる文字は、英数字及び“.”（ドット）であり、漢字は扱えない。また、同一ディレクトリ内においてファイル名、および添付ファイル名については、同一名称のものは扱えない。

4.2.1 送信ファイル

送信するファイルは、設定ファイルの `send_directory` タグで指定されたディレクトリ（送信フォルダ）に格納する。添付ファイルを送信する場合は、対応する送信ファイルの拡張子を除いた同名のフォルダを作成し、その直下に格納する必要がある。

(1) 送信フォルダの例（添付なしの場合）

```
/opt/jeita/ediclient/data/send/  
    00001.xml  
    0002.xml
```

(2) 送信フォルダの例（添付ありの場合）

```
/opt/jeita/ediclient/data/send/  
    00001.xml  
    00001/  
        Parts.gif  
    00002.xml  
    00002/  
        Orderlist.pdf
```

4.2.2 受信ファイル

受信したファイルは、設定ファイルの `receive_directory` タグで指定されたディレクトリ（受信フォルダ）に格納される。添付ファイルを受信した場合は、対応する受信ファイルの拡張子を除いた同名のフォルダが作成され、その直下に格納される。

(1) 受信フォルダの例（添付なしの場合）

```
/opt/jeita/ediclient/data/receive/  
    20191201102254235_receivefile.dat
```

(2) 受信フォルダの例（添付ありの場合）

```
/opt/jeita/ediclinet/data/receive/  
    20191201102254235_receivefile.dat  
    20191201102254235/  
        Attach.pdf  
        Attach2.jpg
```

4.3 ASP から証明書のインストールを依頼された場合

ASP が利用する SSL 証明書によっては、共通クライアントに予めインストールする必要がある。以下、ASP から依頼された場合に作業を行う。

4.3.1 証明書の取得

ASP から依頼されたインストール対象の証明書を取得する。証明書の種類により、ASP から取得（サーバ証明書）する場合と、認証局(CA)の Web サイト等から対象の証明書を取得（root 証明書、中間証明書）する場合がある。

4.3.2 証明書のインストール

Java に同梱されている、keytool を利用してインストールを行う。keytool は、X.509 v1、v2、v3 の証明書、および、PKCS#7 形式の証明書から構成されている PKCS#7 形式の証明書チェーンをサポートしている。インストール作業は管理者権限で作業を行わなければならない。作業はコマンドプロンプトで実行する。

```
(Java インストールパス)/bin/keytool -import -keystore (java インストールパス)/lib/security/cacerts -file (インストール対象証明書)
```

パスワードには、"changeit"を指定する。

インストール対象の情報が出力されるので、内容を確認した後、

“この証明書を信頼しますか? [いいえ]:

の質問に"はい"を入力するとインストールが完了する。

(実行コマンド例)

```
/usr/OpenJDK/jdk-11.0.2/bin/keytool -import -keystore /usr/OpenJDK/jdk-11.0.2/lib/security/cacerts -file /temp/SSL.cer
```

注：インストール対象 (-file) で指定するファイルはフルパスで指定すること。また、（アクセスが拒否されました）となる場合は、書き込み権のあるフォルダに対象ファイルをコピーし、更新した後、元のファイルを上書きすると良い。

5 起動方法

5.1 通信実行

共通クライアントの起動方法は、シェルスクリプトによる実行と API による実行の二通りがある。

5.1.1 シェルスクリプトによる実行

シェルスクリプト (edicient.sh) に引数を与えて実行することにより、共通クライアントを起動する。(通信が行われる) 起動引数で異なる設定ファイルを指定することにより複数の接続先と通信することが可能である。実行は、シェルスクリプトのディレクトリに移動してから行うこと。

(1) コマンド概要

シェルスクリプトによる起動のコマンド形式は以下である。

```
シェルスクリプトファイル名 パラメタ1 パラメタ2 パラメタ3...
```

コマンドは、シェルスクリプトファイル名と複数のパラメタで構成される。シェルスクリプトファイル名とパラメタ、パラメタ同士はスペースで区切り、パラメタの順序は入れ替え可能とする。

また、パラメタは、`-パラメタ名 "パラメタの値"` という形式で設定する。パラメータ名とパラメータ値の間は半角スペースを入れる。パラメタの値はダブルクォーテーションで囲まなくても良いが、パラメタの値にスペースが含まれる場合は、ダブルクォーテーションで囲むことは必須とする。

Pull 型送受信コマンドの戻り値は、プロセスのリターンコードで返す。

(2) 送信コマンド仕様

送信時は以下のコマンドを利用する。

```
./edicient.sh -mode "send" -initfile "path" [-datadir "dir"] [-boxid "boxid"]
```

パラメタ名	必須	候補値	説明
-mode	○	send	動作モード
-initfile	○		設定ファイルのパス
-datadir			送信ファイル・添付ファイルの存在するフォルダのパス
-boxid			ボックス ID 接続先から通知される

表 5.1 送信コマンドパラメタ

-datadir、-boxid は、設定ファイル内にも同様の項目が設定可能であり、引数か設定ファイル内のどちらかで指定しなければならない。

設定ファイル内で指定する場合は、引数は省略可能である。また、引数で指定する場合は、設定ファイル内の設定項目は省略可能とする。両方で指定している場合は、引数が優先される。両方とも指定していない場合は、エラーとする。

戻り値については、「5.2 戻り値とエラー処理方針」を参照。

(3) 受信コマンド仕様

受信時は以下のコマンドを利用する。

```
./ediclient.sh -mode "receive" -initfile "path" [-datadir "dir"] [-boxid "boxid"]
```

パラメタ名	必須	候補値	説明
-mode	○	receive	動作モード
-initfile	○		設定ファイルのパス
-datadir			受信ファイル・添付ファイルを保存するフォルダのパス
-boxid			ボックス ID 接続先から通知される

表 5.2 受信コマンドパラメタ

-datadir、-boxid は、設定ファイル内にも同様の項目が設定可能であり、引数か設定ファイル内のどちらかで指定しなければならない。

設定ファイル内で指定する場合は、引数は省略可能である。また、引数で指定する場合は、設定ファイル内の設定項目は省略可能とする。両方で指定している場合は、引数が優先される。両方とも指定していない場合は、エラーとする。

戻り値については、「5.2 戻り値とエラー処理方針」を参照。

5.1.2 Java API による実行

(1) 送受信 API

jp.jeita.ediclient.core.EDIClient クラスに、送信／受信を行う API が実装されている。

○ PutDocument

クラス : jp.or.jeita.ediclient.core.EDIClient.

メソッド : int PutDocument(String initFilePath, String sendDirPath, String boxId)

○ GetDocument

クラス : jp.or.jeita.ediclient.core.EDIClient.

メソッド : int GetDocument(String initFilePath, String sendDirPath, String boxId)

API 仕様に関しては、「共通クライアント利用解説書付録 1_javadoc」を参照。

(2) 同時多重処理時のログ出力について

API による実行でマルチスレッド処理を行う際は、各処理の設定ファイルで指定するログのレベル・出力先ファイルは統一しなければならない。

5.2 戻り値とエラー処理方針

共通クライアントの戻り値とエラー処理方針は以下の通りである。

戻り値は、シェルスクリプト起動の場合は、コマンドの終了ステータス(\$?)として、API 起動の場合は、メソッドの戻り値(int)として返される。

戻り値	説明	発生可能性		エラー処理方針
		送信	受信	
	正常			
0	正常に処理が終了。	○	○	・正常終了
1	受信可能メッセージが存在しない。		○	・正常終了
20番台	WARNING(正常)			
20	ログ出力失敗。 通信は行われているため正常終了とする。	○	○	・正常終了 ・警告を出力してユーザに通知してもよい
21	送信済みファイル移動不可。 メッセージの送信は行われているため正常終了とする。	○		・正常終了 ・警告を出力してユーザに通知してもよい
22	メッセージ受信したが、ReliabilityAckの送信に失敗した。 サーバ側でメッセージが受信済みになっていないため、重複受信の可能性がある。 メッセージの受信は行われているため正常終了とする。		○	・正常終了 ・警告を出力してユーザに通知してもよい
30番台	失敗(リトライ可能な異常)			
30	サーバに接続不可。 クライアントに問題なく、サーバ側の問題で接続できない可能性があるためリトライ不可とする。	○	○	・時間を置いてリトライ ・APで設定した回数リトライして接続できなければエラーを出力してユーザに通知
40番台	失敗(リトライ不可能な異常)			
40	引数誤り。	○	○	・エラーを出力しユーザに通知
41	環境の異常。 JEITAクライアント内部定義ファイルの異常。	○	○	・エラーを出力しユーザに通知
42	設定ファイル異常。	○	○	・エラーを出力しユーザに通知
43	送受信ファイル入出力時の異常。	○	○	・エラーを出力しユーザに通知
44	送信ファイルサイズオーバー。	○	○	・エラーを出力しユーザに通知
45	ebMSv3送受信機能内部で、通信実行前に発生したエラー。 詳細な原因はログを参照する。	○	○	・エラーを出力しユーザに通知
46	ebMSv3送受信機能内部で、通信実行時に発生したエラー。 詳細な原因はログを参照する。	○	○	・エラーを出力しユーザに通知
47	サーバ接続時に認証に失敗した。	○	○	・エラーを出力しユーザに通知
49	その他のエラー。	○	○	・エラーを出力しユーザに通知

表 5.3 共通クライアントの戻り値とエラー処理方針

5.3 送受信処理

5.3.1 送信処理の動作

(1) ファイル送信順序

送信フォルダに複数のファイルがある場合、ファイル名の昇順でソートした順番で送信される。添付ファイルが複数ある場合も同様に、添付ファイル名の昇順でソートした順番で送信される。

(2) 送信が完了したファイル

- 送信が完了したファイルは、下記の規則に従いファイル名が変更され、送信完了フォルダに移動される。

ファイル名規則：

[日時(年月日時分秒ミリ秒)]_[送信時ファイル名]

例) 20081201102254235_sendfile.dat

- 添付ファイルは、格納フォルダ名が以下の形式に変更され、送信完了フォルダに移動される。(添付ファイル名は変更なし)

フォルダ名規則：

[日時(年月日時分秒ミリ秒)]

例) 20081201102254235

(3) 送信に失敗したファイル

- 送信に失敗したファイルは、下記の規則に従いファイル名が変更され、送信失敗フォルダに移動される。

ファイル名規則：

[日時(年月日時分秒ミリ秒)]_[送信時ファイル名]

例) 20081201102254235_sendfile.dat

- 添付ファイルは、格納フォルダ名が以下の形式に変更され、送信失敗フォルダに移動される。(添付ファイル名は変更なし)

フォルダ名規則：

[日時(年月日時分秒ミリ秒)]

例) 20081201102254235

5.3.2 受信処理の動作

(1) 受信ファイル

- ・受信したファイルは、下記の規則に従いファイル名が変更され、受信フォルダに格納される。

ファイル名規則：

[日時(年月日時分秒ミリ秒)]_[受信ファイル名]

例) 20081201102254235_receivefile.dat

- ・受信した添付ファイルは、受信したファイル名（日時_受信ファイル名）の日時と同一名のフォルダが作成され、その直下に格納される。

フォルダ名規則：

[日時(年月日時分秒ミリ秒)]

例) 20081201102254235

5.3.3 制限事項

(1) 送信ファイルサイズ

共通クライアントで送信できるファイルにはデフォルト設定では以下の制限がある。

1. 送信ファイルのサイズは 100M バイト未満。
2. 送受信ファイル+添付ファイルの合計容量は 200M バイト未満。

上記の制限を越えた場合は、戻り値：44 で異常終了する。

ファイルサイズ制限は、ebMSv3 通信機能定義ファイル (ebmsv3.properties) で変更することができる。

(2) ファイル数

送受信において、添付ファイル数が多いと、以下の問題が発生することがある。このため、送受信時の添付ファイル数は 100 個を目安としていただきたい。

1) 送受信時

ファイル数が多くなると、送受信の処理時間が長くなる。

2) 受信時

ファイル数が多すぎると、受信処理が戻り値 46 で異常終了する。

また、共通クライアントでは、ファイル容量管理は行っていない。このため、長期的な運用においては、受信ディレクトリや送信完了ディレクトリなどに十分な空き領域を用意しておくか、これらのファイルのバックアップ、または消去するなどの運用が必要である。

5.4 ログファイル

5.4.1 ログファイルの種類

ログファイルには、トレースログ、エラーログ、送受信結果一覧ログがある。各ログは、最大 1M バイトで、10 世代でローテーション（※注）を行う。ローテーションしたファイル名はファイル名の後ろに 1~10 の数字が付加される。

例：ログファイルが `trace.log` の場合

(1) 1 世代ローテーションした後の状態

`trace.log`, (ログ出力中のファイル)

`trace.log.1` (ローテーションしたファイル)

(2) 3 世代ローテーションした後の状態

`trace.log`, (ログ出力中のファイル)

`trace.log.1` (ローテーションしたファイル)

`trace.log.2` (ローテーションしたファイル)

`trace.log.3` (ローテーションしたファイル、最も古いファイル)

※注 ログファイルのサイズが 1M を超えた時点で、別ファイルにローテーションするため、ログファイルが 1M を超える場合がある。

5.4.2 トレースログ

動作した履歴が全て出力される。出力する内容は、設定ファイルのログレベルに依存する。ログレベルには、“ERROR”、“Info”、“Debug”の 3 種類がある。詳細は、7 ログ/エラーメッセージ一覧を参照。

デフォルトファイル名：`trace.log`

5.4.3 エラーログ

異常となった動作の履歴が出力される。詳細は、7 ログ/エラーメッセージ一覧を参照。

デフォルトファイル名：`error.log`

5.4.4 送受信結果一覧ログ

通信毎に送信時刻、送受信結果、sid、ボックス ID、ファイルサイズ、エラーコードが出力される。

デフォルトファイル名：`send_recv_result.log`

```
2019-07-18 13:50:28 [NG] SEND sid:170718145926825 No=0001 BoxID=ORDER FileName=order.txt Bytes=7125 Code=30
2019-07-18 14:59:28 [OK] SEND sid:170718145926825 No=0001 BoxID=ORDER FileName=order.txt Bytes=7125 Code=0
```

図 5.1 送受信結果一覧ログ例

5.5 複数接続先と通信を行う場合

複数接続先と通信を行う場合の設定、利用方法について記述する。

5.5.1 設定

接続先 URL、ID が異なるため、接続先ごとに設定ファイル (`initfile.xml`) を用意する。

送受信を行うディレクトリ指定は、同時に起動することを踏まえて、接続先ごとに用意することが望ましい。

5.5.2 送受信

送受信実行時に起動引数(`-initfile`)で接続先の設定を指定し実行する。複数起動することで、同時に異なる接続先への送受信は可能である。起動毎に **Java VM** が起動するため、メモリ等のリソースに注意すること。

6 トラブルシューティング

共通クライアントで発生するエラーについて、以下に示す。各項目は、表 6.1 表記方法の形式で表す。

戻り値	現象
発生原因	
解決方法	

表 6.1 表記方法

6.1 実行環境

—	インストールできない。
・ 共通クライアントのインストール先と同名のフォルダが既に存在している。	
・ 共通クライアントのインストール先と同名のフォルダを削除する。	

9009	共通クライアントが起動しない。
・ Java がインストールされていない。(シェルスクリプト起動のみ)	
・ 「3.1 Java 実行環境のインストール」を参照し、Java をインストールする。	

41、45	共通クライアントの内部定義ファイル異常。
・ 共通クライアントのライブラリの異常。	
・ 共通クライアントライブラリ内に格納されている内部定義ファイルの異常。	
・ 「3 インストール」を参照し、共通クライアントの再インストールを行う。	

6.2 送信／受信共通

20	ログが出力されない。
・ ログファイルが読み取り専用になっている。	
・ ログを出力するフォルダのアクセス許可がない。	
・ 設定ファイルのログ設定を確認する。	
・ ログファイルのパーミッションを確認する。	

30	通信に失敗した。
<ul style="list-style-type: none"> ・ サーバに接続できない。 	
<ul style="list-style-type: none"> ・ ネットワーク接続を確認する。 ・ 設定ファイルに記述した ASP の URL が正しいかを確認する。 ・ 設定ファイルに記述したプロキシ情報が正しいかを確認する。 ・ SSL サーバ証明書、あるいは SSL サーバ証明書の root 証明書がインストールされていることを確認する。 	

40	引数が誤っている。
<ul style="list-style-type: none"> ・ 動作モードが指定されていない／”send”、”receive”以外を指定している。(シェルスクリプト起動のみ) ・ 設定ファイルが指定されていない／設定ファイルのパスが空文字か 201 字以上。 ・ 送信／受信フォルダのパスが空文字か 201 字以上。 ・ 未定義の引数が与えられた。(シェルスクリプト起動のみ) 	
<ul style="list-style-type: none"> ・ 「4 動作設定」を参照し、正しい引数で共通クライアントを起動する。 	

42	設定ファイルが異常。
<ul style="list-style-type: none"> ・ 設定ファイルが誤っている。 	
<ul style="list-style-type: none"> ・ 「4.1 設定ファイル」を参照し、設定ファイルを修正する。 	

43	送受信ファイルの入出力エラー。
<ul style="list-style-type: none"> ・ 指定された送信フォルダが存在しない。 ・ 指定された受信フォルダが存在せず、作成することもできない。 ・ 指定された送信／受信フォルダにアクセスできない。 ・ 受信したファイル／添付ファイルを保存できない。 	
<ul style="list-style-type: none"> ・ 引数／設定ファイルの送信フォルダを修正する。 ・ 送信フォルダを作成する。 ・ 受信フォルダと同名のファイルが存在しているならば削除する。 ・ 送信フォルダ／受信フォルダのパーミッションを書き込み可能にする。 	

46	ASP 側でエラーが発生した。
<ul style="list-style-type: none"> ・ ASP 側でエラーが発生した。 	
<ul style="list-style-type: none"> ・ ASP に問い合わせて原因を確認する。 ・ 送受信ファイルの添付ファイル数が 101 個以上でないか確認する。 	

47	認証エラー
----	-------

<ul style="list-style-type: none"> ・ ユーザ ID、もしくはパスワードが誤っている。
<ul style="list-style-type: none"> ・ 設定ファイルのユーザ ID、パスワードが正しいかを確認する。

6.3 送信処理

21	送信後、送信したファイルが送信完了フォルダ、もしくは送信失敗フォルダに移動されない。
	<ul style="list-style-type: none"> ・ 送信完了フォルダ／送信失敗フォルダにアクセス許可がない。 ・ 送信ファイル／添付ファイルが読み取り専用になっている。 ・ 送信完了フォルダ／送信失敗フォルダのパーミッションを確認する。

44	送信ファイルサイズオーバー
	<ul style="list-style-type: none"> ・ 送信対象ファイルのサイズが 100M バイトを超えている。 ・ 送信対象ファイルと添付ファイルの合計サイズが 200M バイトを超えている。 ・ 複数回に分割して送信する。

6.4 受信処理

22	受信確認通知の送信失敗
	<ul style="list-style-type: none"> ・ 受信処理中に通信が切断された。 ・ 受信処理中に ASP 側でエラーが発生した。 ・ ネットワーク接続を確認する。 ・ ASP に問い合わせて原因を確認する。

7 ログ/エラーメッセージ一覧

エラーログ/トレースログに出力される主なメッセージの一覧を以下に示す。

7.1 ERROR レベルのログに出力されるメッセージ

メッセージ	説明
Communication error occurred.	通信異常が発生した場合に表示される。
edicient.properties cannot be opened.	共通クライアントのコアライブラリ (jeita-edicient.jar)内に含まれる定義ファイルを開くことができない。
edicient.properties does not exist.	
It failed in certification.	WSS ユーザ認証に失敗した。
It failed in sending and receiving functional initialization process.	通信機能の初期化に失敗した。
It failed in the class load of AdaptorController.	共通クライアントの ebMSv3 通信アダプタライブラリ (jeita-edicient-adaptor-ebmsv3.jar) が壊れている。
It failed in the class load of TransferFactory.	
It failed in the notification of the receiving completion.	受信処理時、サーバへの受信完了通知の送信に失敗した。(戻り値 2 2)
It is the unexpected return code.	通信機能で想定外のエラーが発生した。
The size of the sending file has exceeded Max byte.	送信ファイルのサイズが最大ファイルサイズをこえている。
The folder to save the attachment file cannot be created.	受信処理時、受信した添付ファイルを保存するフォルダを作成できない。
The received attachment file cannot be written in.	受信処理時、受信した添付ファイルを保存できない。
The receiving folder could not be created.	受信処理時、受信フォルダを作成できない。
The schemer file of the setting file is not found.	共通クライアントのコアライブラリ (jeita-edicient.jar)内に含まれる設定ファイルのXMLスキーマを開くことができない。
The sending file cannot be read.	送信処理時、送信対象のファイルを開くことができない。
The sending file or/and attachment file cannot be moved to the sending completion folder.	送信処理の成功時、送信したファイル/添付ファイルを送信完了フォルダに移動することができない。
The sending file or/and attachment	送信処理の失敗時、送信を試みて失敗したファ

t file cannot be moved to the sending failure folder.	イル／添付ファイルを送信失敗フォルダに移動することができない。
The sending folder cannot be accessed.	送信フォルダにアクセスすることができない。
The sending folder does not exist.	送信フォルダが存在しない。
Total size of the sending file and the attachment file has exceeded Max byte.	送信ファイルと添付ファイルの合計サイズが最大ファイルサイズをこえている。
You cannot connect to the server.	サーバに接続できない。

7.2 Info レベルのログに出力されるメッセージ

○:任意の文字列

メッセージ	説明
Any sending file does not exist.	送信処理を開始したが、送信フォルダが空である。
Process ends. Return value:○	共通クライアントの処理が終了した。戻り値は○である。
Receiving end was notified by transmission function. : ○	受信処理のための通信が終了した。通信機能からの戻り値は○である。
Receiving process is indicated to transmission function.	通信機能に対して、受信開始を指示した。
Receiving process starts.	受信処理を開始した。
Send end was notified by transmission function. : ○	送信処理のための通信が終了した。通信機能からの戻り値は○である。
Send process is indicated to transmission function.	通信機能に対して、送信開始を指示した。
Sending process starts.	送信処理を開始した。
The attachment file is moved to the sending completion folder. Folder name after the moving: ○	送信した添付ファイルを送信完了フォルダ (○) へ移動した。
The attachment file is moved to the sending failure folder. Folder name after the moving: ○	送信を試みて失敗した添付ファイルを送信失敗フォルダ (○) へ移動した。
The attachment file is read.	送信のため、添付ファイルを読み込んだ。
The attachment file is written.	受信した添付ファイルを保存した。
The receiving file is written.	受信したファイルを保存した。
The sending file is moved to the s	送信したファイルを送信完了フォルダ (○) へ

ending completion folder. File name after the moving: ○	移動した。
The sending file is moved to the sending failure folder. File name after the moving: ○	送信を試みて失敗したファイルを送信失敗フォルダ (○) へ移動した。
The sending file is read.	送信のため、送信対象ファイルを読み込んだ。
The standby is done until receives.	受信のための通信が終了するまで待機する。
The standby is done until sending ends.	送信のための通信が終了するまで待機する。

7.3 Debug レベルのログに出力されるメッセージ

○:任意の文字列

メッセージ	説明
Argument analytical result. Operational mode: ○ Setting file: ○ Sending and receiving folder: ○ BoxID: ○	共通クライアントに与えられた引数。
Attachment file name: ○ Attachment file size: ○	送信する添付ファイルのファイル名とサイズ。
Communication setting information. operational mode: ○ UserID: ○ BoxID: ○	通信機能に渡す通信設定情報。
Communication setting superscription information. BoxID: ○ UserID: ○ Password: ○ EndPoint: ○	通信機能に渡す通信設定上書き情報。
Receiving file name: ○	受信したファイルに付与したファイル名。
Receiving process result return value: ○ Message: ○	受信処理が終了した。その時の戻り値と通信機能からのメッセージ。
Return code of reception process from transmission function: ○	受信のための通信が終了した。その時の通信機能からの戻り値。
Return code of sending process from transmission function: ○	送信のための通信が終了した。その時の通信機能からの戻り値。
Sending file name: ○ Sending file size: ○	送信対象のファイル名とサイズ。
Sending process result return value: ○ Message: ○	送信処理が終了した。その時の戻り値と通信機能からのメッセージ。

Setting file analytical result. UserID: <input type="radio"/> Password: <input type="radio"/> URL: <input type="radio"/> Proxy host: <input type="radio"/> Proxy port: <input type="radio"/> Proxy use rID: <input type="radio"/> Proxy password: <input type="radio"/> Sending folder: <input type="radio"/> Sending completion folder: <input type="radio"/> Sending failure folder: <input type="radio"/> Receiving folder: <input type="radio"/> The number of receiving maximum messages: <input type="radio"/> Re-receiving folder: <input type="radio"/> Sending BoxID: <input type="radio"/> Receiving BoxID: <input type="radio"/> Designated receiving BoxID: <input type="radio"/> Error log file: <input type="radio"/> Trace log file: <input type="radio"/> Log level: <input type="radio"/> Query: <input type="radio"/>	与えられた設定ファイルの解析結果。
The end of the receiving process was notified. Transaction ID: <input type="radio"/>	受信のための通信が終了したことを、通信機能から通知された。 <input type="radio"/> は終了した通信のトランザクションID。
The end of the sending process was notified. Transaction ID: <input type="radio"/>	送信のための通信が終了したことを、通信機能から通知された。 <input type="radio"/> は終了した通信のトランザクションID。
The receiving file of the same date was checked. File name: <input type="radio"/>	同日付で受信済みのファイル (<input type="radio"/>)を確認。
The receiving folder is created. Folder name: <input type="radio"/>	受信フォルダ (<input type="radio"/>)を作成した。
The receiving folder was decided. : <input type="radio"/>	受信フォルダを <input type="radio"/> に決定した。
The sending folder was decided. : <input type="radio"/>	送信フォルダを <input type="radio"/> に決定した。
The wait for receiving starts. Transaction ID: <input type="radio"/>	受信のための通信が終了するまで待機する。 <input type="radio"/> は終了待ちの通信のトランザクションID。
The wait for sending starts. Transaction ID: <input type="radio"/>	送信のための通信が終了するまで待機する。 <input type="radio"/> は終了待ちの通信のトランザクションID。
Total capacity of sending file and attachment file: <input type="radio"/>	送信対象ファイルと添付ファイルのサイズの合計が <input type="radio"/> である。