VeriTrans4G 開発ガイド（MDK モジュール方式）

Ver. 1.0.11 （2026 年 10 月～）

Copyright © 2026 DG Financial Technology, Inc., a Digital Garage company. All rights reserved.

目次

第 1 章　導入の前に 3
1-1 本ガイドの内容 3
1.1.1 MDK（Merchant Development Kit）について 3
1-2 著作権、および問い合わせ先 3
1-3 改訂履歴 3

第 2 章　サービス概要 5
2-1 利用可能な決済手段 5
2-2 ワンクリック継続課金サービスについて 5
2-3 不正検知サービスについて 6

第 3 章　MDK（Merchant Development Kit）の導入 7
3-1 MDK の導入手順 7
3-2 利用可能な MDK コマンド 8
3-3 通信エラー発生時の注意点 9

第 4 章　インターフェース詳細 11
4-1 決済インターフェース詳細 11
4.1.1 各サービスのインターフェース詳細 11
4.1.2 共通補助パラメータ 12
4-2 決済結果通知 13
4.2.1 電文仕様 13
4.2.2 各決済の結果通知一覧 14
4.2.3 結果通知電文 インターフェース詳細 15
4.2.4 HMAC による結果通知の改ざんチェック 15

第 5 章　決済テスト 17
5-1 テスト実施にあたっての注意事項 17
5-2 MAP を利用したテスト取引の結果確認 17

第1章 導入の前に

1-1 本ガイドの内容

本ガイドは、株式会社 DG フィナンシャルテクノロジーが提供する VeriTrans4G を利用するための専用ソフトウェア MDK（Merchant Development Kit）をインターネット店舗などに導入する開発者向けのガイドです。

開発の際の参考となるサンプルや、提供ライブラリの説明、電文フォーマットなどについて記載しています。

1.1.1 MDK（Merchant Development Kit）について

MDK は、店舗様サイトに組み込んで頂く、ライブラリモジュールの総称です。Java、PHP 等　複数言語の MDK をご用意しております。MDK のサポートページからサーバー環境に合った MDK をダウンロードして下さい。

店舗様サイトに組み込み、設定・カスタマイズを行うことによって、ご契約のサービスで決済を行うことができます。複数のサービスを同時に利用可能です。

✓ **動作環境については別冊『MDK インストールガイド』をご参照ください。**

1-2 著作権、および問い合わせ先

［著作権］
本ドキュメントの著作権は株式会社 DG フィナンシャルテクノロジーが保有しています。
Copyright © 2026 DG Financial Technology, Inc., a Digital Garage company. All rights reserved.

［お問い合わせ先］
株式会社 DG フィナンシャルテクノロジー　ベリトランス　テクニカルサポート
電子メール：　tech-support@veritrans.jp

1-3 改訂履歴

2017/04　：Ver1.0.0 リリース
2017/07　：Ver1.0.1 リリース
「3-3　通信エラー発生時の注意点」を追加
「4.2.2　各決済の結果通知一覧」の「その他」の列に、PayPal の確報通知（売上・返金）の記載を追加
2017/08　：Ver1.0.2 リリース
「4.2.1　電文仕様」の　content-hmac　に関する説明を追記
2018/08　：Ver1.0.3 リリース
「4.2.2　各決済の結果通知一覧」の「Alipay」の「返金」列の誤りを修正
誤：　◎申込結果通知
正：　△申込結果通知
2018/09　：Ver1.0.4 リリース
「4.1.2　共通補助パラメータ」の「取引メモ 1」の書式・制限を修正
2018/12　：Ver1.0.5 リリース
楽天ペイの表記を楽天ペイ（オンライン決済）に変更
「3-2　利用可能な MDK コマンド」に Alipay の Confirm を追加

開発ガイド（MDK モジュール方式）

2019/5 ：Ver1.0.6 リリース

「4.2.2 各決済の結果通知一覧」の修正

アプリ利用時の結果通知に関する説明を追記

2023/9 ：Ver1.0.7 リリース

MasterPass を削除（サービス終了のため）

永久不滅ポイント決済を削除（サービス終了のため）

「MPI ホスティング」を「3D セキュア」に名称変更

「4.2.2 各決済の結果通知一覧」にファミペイ、AmazonPay を追加

2024/03 ：Ver1.0.8 リリース

「5-2 MAP を利用したテスト取引の結果確認」MAP の URL を pay.veritrans.co.jp から admin3.veritrans.co.jp に変更

2024/11 ：Ver1.0.9 リリース

電子マネー決済から nanaco を削除（サービス終了のため）

2025/10 ：Ver1.0.10 リリース

「2-1 利用可能な決済手段」 キャリア決済の種類を修正

「4.2.1 電文仕様」「マーチャント秘密鍵」を「マーチャント認証鍵」に変更

「4.2.1 電文仕様」の「結果通知受信処理の結果について」の説明を修正

2026/10 ：Ver1.0.11 リリース

「4.2.4 HMAC による結果通知の改ざんチェック」に検証を追加

LINE Pay の記述を削除

リクルートかんたん支払い の記述を削除

第2章 サービス概要

2-1 利用可能な決済手段

利用可能な決済手段は下表のとおりです。

表 2-1-1 決済サービス一覧

決済サービス名	説明
クレジットカード決済	各種ブランド、カード会社等発行のクレジットカードでのお支払いが可能です。
3D セキュア	クレジットカード決済と連動する、3-D Secure による認証サービスです。
コンビニ決済	セブン－イレブン、ローソン、ファミリーマート等のコンビニエンスストア店舗でのお支払いが可能です。
電子マネー決済	電子マネー（Suica、Edy、WAON）を利用した EC でのお支払いが可能です。
銀行決済	銀行、信用金庫等の金融機関の ATM での支払い、および、ネットバンキングでのお支払いが可能です。
銀聯ネット決済（UPOP）	銀聯カードでのお支払いが可能なオンラインサービスです。
PayPal 決済	PayPal のアカウント（メールアドレス）によるお支払いが可能です。
Alipay 決済	Alipay のアカウントを利用したお支払いが可能です。EC 向けのオンライン決済だけでなく、リアル店舗でのバーコードを利用した決済が可能です。
キャリア決済	通信キャリアの通信料金と合算して支払うことができるサービスです。 4 つの通信キャリア（ドコモケータイ払い、au PAY（au かんたん決済）／au PAY（ネット支払い）、ソフトバンクまとめて支払い（B）／ソフトバンクまとめて支払い（A））による支払いが可能です。
ショッピングクレジット決済	ショッピングクレジットによる決済が利用できるサービスです。 Orico Web クレジットによる支払いが可能です。
楽天ペイ（オンライン決済）	楽天会員 ID を利用して決済できるサービスです。 楽天ポイントを利用したお支払いが可能です。

✓ **各決済のシステムフローは、『開発ガイド別冊 概略システムフロー図』をご参照ください。**

2-2 ワンクリック継続課金サービスについて

会員管理機能を利用することで、店舗様が運営する EC サイト等の利用会員（消費者）と、決済に利用するクレジットカード情報、取引結果等を関連付けて管理することが可能です。この「会員管理機能」と「継続課金機能」を組み合わせることで、定期購入の代金やサービスの月額会費等の支払いのための、定期的な決済を行うことができます。

これらのサービスの総称を「ワンクリック継続課金サービス」と呼びます。

ワンクリック継続課金サービスを導入することで、以下の機能をご利用になれます。

- 会員管理（ID 管理）

  サイトの利用会員（消費者）の ID と、クレジットカード情報、継続課金のスケジュールを登録します。

  登録した情報は変更（更新）、削除が可能です。

- 会員 ID を利用した決済

  カード決済要求時に会員 ID を指定し、会員 ID に紐付いたクレジットカード情報で決済します。

  カード以外の決済時にも、会員 ID を指定することにより、決済結果と会員 ID を関連付けることができます。

- 継続課金

  定期購入の代金やサービスの月額会費等の支払いのために、定期的に決済を実行します。

- クレジットカード洗替

  クレジットカード番号や有効期限の有効性を確認し、新しい情報に更新します。

✓ **詳しくは、別冊『ワンクリック継続課金サービス ご利用ガイド』を参照して下さい。**

2-3 不正検知サービスについて

従来のクレジットカード決済に不正検知機能を統合した「完全一体型」のサービスで、店舗様の不正対策業務を支援いたします。店舗様のニーズに合わせて、複数の不正検知エンジンの中から利用するエンジンを選択可能です。

- カード会社への与信要求と並行して不正判定エンジンによる解析を実行し、与信結果（成功/失敗）と不正判定結果（許可/保留/拒否）を併せて店舗様のシステムに返戻します。
- 不正検知の判定結果は、管理ツール（MAP）の取引情報画面で確認できます。

✓ **不正検知サービスのご利用には、別途お申込みが必要です。詳しくは、別冊『不正検知サービス ご利用ガイド』をご参照のうえ、弊社の営業担当までお問い合わせください。**

第3章 MDK（Merchant Development Kit）の導入

3-1 MDK の導入手順

MDK の導入手順について説明します。

No.	作業	作業内容	備考
1	MDK の入手	ベリトランスのサポートページより、店舗様の導入環境に適応する MDK と、関連するドキュメント類をダウンロードします。 https://www.veritrans.co.jp/trial/login/	
2	外部リソースの入手	MDK の導入環境情報を参照し、導入に必要なソフトウェアやライブラリをダウンロード／入手します。	『MDK インストールガイド』を参照して下さい。
3	外部リソースの配置・環境の設定	ダウンロードしたソフトウェアやライブラリの配置・設定を行います。	
4	MDK のインストール	ダウンロードしたファイルを解凍し、インストールします。 認証のために必要な API 設定情報を管理ツール（MAP）より入手し、MDK 設定ファイルに必要な設定を行います。	
5	通信テスト（テスト環境）	ベリトランス折り返しの通信テストを実施します。 ダミーモードを“ON”に設定し、MDK の動作確認、ベリトランスと通信が可能かを確認します。テスト専用アカウントの配布も可能です。	『導入テストガイド』を参照して下さい。
6	通信テスト（本番環境）	各決済センターまでの通信テストを実施します。 ダミーモードを“OFF”に設定し、本番の決済が可能か確認します。 ※この通信は本番取引としてトランザクション料が課金されますのでご注意ください。	

※ 開発言語ごとに MDK が動作する実行環境の要件が定められています。詳細は、弊社ホームページ、または**『MDK インストールガイド』をご参照ください**。

※ 弊社動作確認済み環境に合致しない環境（OS 等）に導入される場合にはサポートを致しかねる場合がございます。
また、以下の項目については原則としてサポートは致しかねますので、予めご了承ください。

- 独自にカスタマイズされた OS（カーネル）上での動作確認
- 独自にカスタマイズされたプログラムに関するお問い合わせ
- MDK とは関連のないサーバー環境設定等

3-2 利用可能な MDK コマンド

表 3-2-1 利用可能な MDK コマンド

コマンド名		Authorize	ReAuthorize	Capture	Cancel	Refund	その他	Search
決済サービス名	クレジットカード決済	○ 与信	○ 再取引	○ 売上	○ キャンセル			○ 検索
	3D セキュア	○ 本人認証	○ 再取引	※カード決済の売上、キャンセルを実行します。				○ 検索
	コンビニ決済	○ 申込		△ （入金通知）	○ キャンセル			○ 検索
	電子マネー決済	○ 申込		△ （入金通知）	○ キャンセル	○ 返金		○ 検索
	銀行決済	○ 申込		△ （入金通知）				○ 検索
	銀聯ネット決済（UPOP）	○ 与信		○ 売上	○ キャンセル	○ 返金		○ 検索
	PayPal 決済	○ 与信		○ 売上	○ キャンセル	○ 返金		○ 検索
	Alipay 決済	○ 与信同時売上				○ 返金／返金申込	**Confirm** 決済ステータス確認	○ 検索
	キャリア決済	○ 与信または継続課金申込		○ 売上	○ キャンセル		**Terminate** 継続課金終了	○ 検索
	ショッピングクレジット決済	○ 画面表示		△ （審査結果通知）				○ 検索
	楽天ペイ（オンライン決済）	○ 申込		○ 売上	○ キャンセル			○ 検索
説明		与信や、支払の申込を要求します。	過去の取引情報を元に再度与信、や支払の申込を要求します。	与信済みの取引の売上確定を要求します。	与信・申込や売上に対する取消を行います。	売上済み取引に対し返金を要求します。	各サービスの『インターフェース詳細』をご参照ください。	取引情報の検索を行います。

✓ **サービス毎に、コマンドの利用条件は異なりますので、各サービスの『インターフェース詳細』をご参照ください。**

◆ Authorize コマンド（与信、申込）

通常、決済の申込のために最初に実行するコマンドは、Authorize です。Authorize に失敗した場合は、決済を完了させることができませんので、消費者には決済が失敗したことをご案内し、決済の申込をやり直して頂く必要があります。

◆ Capture コマンド（売上）

Authorize コマンドで与信や申込が成立した場合、Capture コマンドで売上を確定する必要があります。売上を確定しないと、消費者への請求対象にはなりません。ただし、以下のケースを除きます。

- Authorize コマンド要求時に、「同時売上」のオプションを指定した場合は、Authorize コマンドが成功したタイミングで売上が確定します。（クレジットカード決済等）

- 表 3-2-1 の Capture コマンドの列に、△が記入されている決済サービス（コンビニ決済、電子マネー決済など）では、消費者の入金等のアクションにより、決済事業者が売上確定の処理を実施したタイミングで売上が確定します。この場合は、決済サーバーから店舗様システムに売上が確定したことを示す通知を送信します。
- キャリア決済の継続課金など、弊社または決済事業者側の処理で自動的に売上が成立するケースでは、売上を要求する必要はありません。

◆ Cancel／Refund コマンド（キャンセルおよび返金）

決済のキャンセル（取消）または返金を実行する場合、Cancel コマンドと Refund コマンドの2つのコマンドを利用するサービスでは、各サービスの仕様に従い、Cancel コマンド（取消）と Refund コマンド（返金）を使い分ける必要があります。

Cancel コマンドのみを利用するサービスでは、コマンド要求時の取引の状態によって、決済サーバーまたは決済事業者側の判断で取消または返品を適切に処理しますので、店舗様システムでは取消か返金かを意識する必要はありません。

◆ ReAuthorize コマンド（再取引）

クレジットカード決済（3D セキュアを含む）では、過去の取引で利用したクレジットカード情報を利用して、与信（および与信同時売上）を再実行可能です。要求時には、過去の取引 ID（元取引 ID）をパラメータに指定します。

再取引は、ベリトランスに取引履歴が保管されている間は実行可能です。保管期間は、取引のステータスが最後に変化した日を起算日として 400 日間を標準とします。

以下に、再取引可能期間を示します。

取引ステータス	再取引可能期間 ※取引履歴の保管期間
売上	売上実行日+400 日
取消（返品）	取消（返品）実行日＋400 日
与信（与信有効期限切れ）	与信日＋与信有効期間（60 日）＋400 日

保管期間を超えると取引履歴は削除されますので、再取引に利用する元取引 ID は、直近の再取引に使用した取引 ID を指定することをお勧めします。

◆ Search コマンド（検索）

取引の状態を検索することが可能です。

（重要）

検索条件は、決済サービス毎に様々な条件を指定可能ですが、原則として必ず「取引 ID」を条件に含めた検索を行って頂きますようお願いします。取引 ID を条件に含まない検索を実施された場合、データ量によってはレスポンス遅延や、想定外の結果が返戻される場合がございます。取引 ID を含まない、複雑な条件での検索を実施される場合は、事前にテクニカルサポートまでご相談下さい。

3-3 通信エラー発生時の注意点

MDK と弊社サーバー間のネットワーク障害等で通信異常によるエラーが発生した場合、MDK は先頭が「M」で始まる結果コードを返します。このとき、決済要求が弊社サーバーに到達している可能性がありますので、該当の決済が成立しているか否かは、管理画面（MAP）の取引検索や MDK の Search コマンド（検索）を利用してご確認ください。

- クレジットカード決済の与信同時売上のように、1 回の要求で決済が成立する決済手段をご利用の場合は、消費者への請求が発生する可能性があるため、必ずご確認ください。
- 後払い型の決済手段（コンビニ決済や銀行決済など）については、決済の申込要求が成功しただけでは入金は成立しませんので、ご確認は必須ではございません。

- ✧ 決済成立の可能性がある通信エラー関連の結果コードは次のとおりです。（2017 年 7 月現在）

  MA99、MB99、MB03、MF01、MF02、MF03、MF04、MF99

  結果コードは今後新しく追加される可能性がありますので、「M」で始まるエラーの発生を検知し、決済成否を確認して頂くことをお勧めします。

- ✧ 結果コードの詳細は、別冊の『結果コード一覧』をご参照下さい。

第4章 インターフェース詳細

4-1 決済インターフェース詳細

4.1.1 各サービスのインターフェース詳細

インターフェース詳細は、決済サービス毎の別冊ドキュメントを参照して下さい。

下表にファイル名の一覧を示します。

表 4-1-1 インターフェース詳細ファイル名一覧

決済サービス名		インターフェース詳細
クレジットカード決済	カード	VeriTrans4G_Development_Guide_IF02_Card_XXX.pdf
	3D セキュア	VeriTrans4G_Development_Guide_ex_3DS2_XXX.pdf
	MDK トークン※1	VeriTrans4G-MDKToken_Development_Guide_XXX.docx
コンビニ決済		VeriTrans4G_Development_Guide_IF04_CVS_XXX.pdf
電子マネー決済		VeriTrans4G_Development_Guide_IF05_EM_XXX.pdf
銀行決済		VeriTrans4G_Development_Guide_IF06_Bank_XXX.pdf
銀聯ネット決済（UPOP）		VeriTrans4G_Development_Guide_IF07_UPOP_XXX.pdf
PayPal 決済		VeriTrans4G_Development_Guide_IF08_Paypal_XXX.pdf
Alipay 決済		VeriTrans4G_Development_Guide_IF10_Alipay_XXX.pdf
キャリア決済		VeriTrans4G_Development_Guide_IF11_Carrier_XXX.pdf
ショッピングクレジット決済（オリコ）		VeriTrans4G_Development_Guide_IF12_Oricosc_XXX.pdf
楽天ペイ（オンライン決済）		VeriTrans4G_Development_Guide_ex_rakuten_XXX.pdf
検索機能		VeriTrans4G_Development_Guide_IF13_Search_XXX.pdf
ワンクリック継続課金サービス	ご利用ガイド	VeriTrans4G_PayNowID_Overview_XXX.pdf
	I/F 詳細	VeriTrans4G_Development_Guide_IF01_PayNowID_XXX.pdf
	I/F 詳細（会員管理）	VeriTrans4G_Development_Guide_IF01-2_PayNowID(会員管理機能).xlsx
	運用ガイド	VeriTrans4G_PayNowID_Operation_Guide_XXX.pdf

注） ファイル名の「XXX」は、バージョンを示す数字です。

※1 経済産業省「クレジット取引セキュリティ対策協議会」より公表されている「クレジットカード取引におけるセキュリティ対策の強化に向けた実行計画」に従い、クレジットカード情報の非保持（非通過および非保持）に対応するためには、決済サーバーへの決済要求時に、トークン化したカード情報（MDK トークン）を利用する必要があります。

4.1.2 共通補助パラメータ

要求電文				
フィールド名	項目名	書式・制限	説明	設定
memo1	取引メモ 1	半角英数字 100 文字以内	取引に関するメモを指定します。	任意
freeKey	キー情報	半角英数字 256 桁以内	店舗様システムで利用している任意のキー情報を設定します。	任意

全ての決済サービスで設定可能なパラメータです。それぞれ以下のような利用を想定しています。

■memo1（取引メモ 1）について

取引に関するメモ情報です。

memo1（取引メモ 1）は、ログ出力の際にマスクされず、入力された値がそのままログファイルに出力されますので、カード番号、個人情報、および機密情報などを入力されないようお願いいたします。

■freeKey（キー情報）について

取引に関するキー情報です。店舗様で別途管理している ID と弊社取引を紐付けるためのキー情報を設定することができます。

（例： 店舗様側システムの取引 ID と弊社取引を紐付ける）

freeKey（キー情報）は、ログ出力の際にマスクされず、入力された値がそのままログファイルに出力されますので、カード番号、個人情報、および機密情報などを入力されないようお願いいたします。

■memo1（取引メモ 1）、freeKey（キー情報）の上書きについて

取引メモ 1、キー情報の値は、最新のトランザクションにて指定した値で上書きされます。

最新のトランザクションで指定した取引メモ 1、キー情報の値のみが保存され、以前のトランザクションで指定した取引メモ 1、キー情報の値は保存されません。

例 1：与信／決済申込（Authorize）時、売上時とも指定した場合

⇒保存されている取引メモ 1、キー情報の値は、売上時に設定した値です。与信／決済申込（Authorize）時に設定した値は保存されません。

例 2：与信／決済申込（Authorize）時は未設定、売上時に設定した場合

⇒保存されている取引メモ 1、キー情報の値は、売上時に設定した値です。

例 3：与信／決済申込（Authorize）時に設定、売上時は未設定の場合

⇒保存されている取引メモ 1、キー情報の値は、与信／決済申込（Authorize）時に指定した値です。

例 4：売上時に設定、その後キャンセルを実施時にも設定した場合

⇒保存されている取引メモ 1、キー情報の値は、キャンセル時に指定した値です。売上時に設定した値は、保存されません。

4-2 決済結果通知

コンビニ決済や銀行決済のように、申込を行ってから決済が完了するまでにタイムラグが発生するサービスでは、結果の確定後に、決済サーバーから店舗様システムに向けて決済結果通知を送信します。

4.2.1 電文仕様

決済結果通知の電文仕様は以下の通りです。

プロトコル		HTTP1.0 または HTTP1.1
メソッド		POST
リクエストヘッダ	**content-hmac**	改ざんチェック用の HMAC 値 設定値： h={アルゴリズム名};s={CCID};v={HMAC 値} ・アルゴリズム名には”HmacSHA256”が設定されます。 ・CCID には、「マーチャント CCID」が設定されます。 ・HMAC 値は「MessageBody」を“h=”に設定されたアルゴリズムでハッシュ化した値です。 ・HMAC 算出時の鍵は、「マーチャント認証鍵」です。
	content-length	BODY 長
	Content-Type	application/x-www-form-urlencoded
ポート番号		443 または 80
文字コード		UTF-8
MessageBody		書式は以下のとおりです。 電文構成要素 1=[値 1]&電文構成要素 2=[値 2]&電文構成要素 3=[値 3]&･･･ （例）コンビニ決済の入金通知電文 numberOfNotify=2&pushTime=20170330172650&pushId=00000237& orderId0000=ORDER001&cvsType0000=sej&amount0000=100&receiptNo0000=0000123& receiptDate0000=20170330172505&rcvAmount0000=100&dummy0000=0& orderId0001=ORDER002&cvsType0001=sej&amount0001=100&receiptNo0001=0000123& receiptDate0001=20170330172505&rcvAmount0001=100&dummy0001=0 ✓ 電文構成要素の詳細は、各サービスの『インターフェース詳細』をご参照ください。

※ 本ドキュメントで規定しない仕様については原則として RFC1945, RFC2616 に準ずるものとします。

■ 結果通知受信処理の結果について

決済サーバーからの結果通知に対して店舗様のシステムが HTTP ステータスコード 200 番台を返戻した場合、決済サーバーは店舗様のシステムの処理が正常に終了したものと判断します。HTTP ステータスコード 200 番台以外のコードを返戻した場合は、受信失敗と判断し、一定期間通知を繰り返します。

- 通常は HTTP ステータスコード”200”を返戻してください。
- 再送は、1～10 分間隔で最長 8 時間試行されます。
- 規定の回数失敗した場合は、通知処理が停止します。受信できなかった取引のステータス確認には、管理ツール（MAP）または Search 機能をご利用ください。

注意事項：

HTTP ステータスコード 200 番台以外を返戻する際、HTTP リダイレクト(3xx の要求)のコードは返却しないようにしてください。

3xx が返戻された場合、リダイレクト先へのアクセスが行われますが、正常にPOSTすることはできません。また、リトライも行われなくなる可能性があります。

■　pushId（識別 ID）について

pushId（識別 ID）は1つの決済サービス内ではユニークですが、複数の決済サービスをご利用の場合、サービスをまたいで同じ ID が設定される場合がありますので、ユニークキーとして処理しないようにご注意下さい。

4.2.2 各決済の結果通知一覧

決済サービス名	与信	売上・入金	キャンセル	返金	その他
クレジットカード決済	―	―	―	―	―
3D セキュア	●決済結果通知	―	―	―	―
コンビニ決済	―	○入金通知	―	―	―
電子マネー決済	―	○入金通知	―	○返金通知	―
銀行決済	―	○入金通知	―	―	―
銀聯ネット決済（UPOP）	●決済結果通知	◎決済結果通知	◎決済結果通知	◎決済結果通知	―
PayPal 決済	―	○入金通知	―	○返金通知	◎確報通知 （売上・返金）
Alipay 決済					
オンライン決済	●決済結果通知	―	―	△申込結果通知 ◎決済結果通知	―
バーコード決済 （店舗スキャン型）	―	―	―	―	―
バーコード決済 （消費者スキャン型）	●決済結果通知	―	―	―	―
キャリア決済	○申込完了通知	―	―	―	○ 継続終了完了通知 ◎ 継続課金結果通知 ○ 継続課金抹消通知
ショッピングクレジット決済	―	―	―	―	◎審査結果通知
楽天ペイ（オンライン決済）	○申込完了通知	◎売上結果通知	◎取消結果通知	―	―
ファミペイ	○決済結果通知				
Amazon Pay	◎決済申込結果通知	◎売上結果通知	◎キャンセル結果通知	◎返金結果通知	◎再与信結果通知 ◎解約結果通知 抹消通知(*1)
ワンクリック継続課金サービス	―	△課金結果通知	―	―	△洗替結果通知

※　表中の○は、結果が成功の場合に送信されます。

※　表中の●は、結果（成功/失敗）に関わらず送信されます。ただし、消費者のブラウザを介するサービスでは、消費者が途中で離脱し決済に進まなかった等のケースで、通知が送信されないことはございます。また、アプリを利用したサービスでは、消費者がアプリを途中で閉じた場合は、通知は送信されません。

※　表中の◎は、結果（成功/失敗）に関わらず送信されます。

※　表中の△は、結果が失敗の場合に送信されます。

※　表中の(*1)は、消費者もしくは事業者により Amazon Pay の随時決済状態が終了となった場合に、失敗相当の結果で送信されます。

4.2.3 結果通知電文 インターフェース詳細

結果通知電文のインターフェース詳細は、決済サービス毎の別冊ドキュメント（表 4-1-1）を参照して下さい。

4.2.4 HMAC による結果通知の改ざんチェック

決済サーバーから送信される決済結果通知電文に対し、HMAC（Hash-based Message Authentication Code）を用いた改ざんチェックを行うことができます。HMAC による検証は、データの完全性および送信元の真正性を確認するための一般的に知られた検証技術です。

第三者による電文の改ざんやなりすましを防ぎ、安全に受信処理を行うため、結果通知受信時には HMAC 値の検証処理を実施することを推奨いたします。なお、MDK をご利用の場合は検証用メソッドをご用意しておりますので、そちらのメソッドのご利用を推奨いたします。
（.各 MDK のメソッド名 Java：checkMessage、NET：CheckMessage、PHP：checkMessage、Ruby：check_message、JavaScript：checkMessage、Go：CheckMessage）

■ 検証時の注意点および手順

1.マーチャント認証鍵の利用

HMAC 算出に使用する鍵には、管理ツール（MAP）より取得した「マーチャント認証鍵」を利用します。

2.認証鍵のバイナリ変換

HMAC の算出にあたり、設定された「マーチャント認証鍵」の文字列はそのまま使用せず、必ず一度バイナリデータ（16 進数デコードによるバイト配列化）へ変換した上で処理を行ってください。

3.HMAC 値の照合

受信したリクエストの MessageBody（受信電文のパラメータ文字列）とバイナリ化したマーチャント認証鍵を用いて HMAC-SHA256 を算出し、リクエストヘッダ content-hmac に含まれる HMAC 値と照合・検証を行います。

■ サンプルコード（JavaScript / Node.js）

以下に、Node.js（JavaScript）を利用して HMAC による改ざんチェックを行うサンプルコードを示します。

```
const crypto = require('crypto')

// MAP でのマーチャント認証鍵を下記(secretkey)に設定
const secretKey = 'xxxx';
const payload = 'numberOfNotify=1&….';
const responseHMAC = 'xxxx';

// 認証鍵（16 進数文字列）をバイナリ化
const binKey = Buffer.from(secretKey, "hex");
const hmac = crypto.createHmac('sha256', binKey);
hmac.update(payload, 'utf8');

const calcHmac = hmac.digest('hex')

// 文字列を Buffer に変換
const calcHmacBuffer = Buffer.from(calcHmac, 'hex');
const responseHmacBuffer = Buffer.from(responseHMAC, 'hex');

if (
  calcHmacBuffer.length === responseHmacBuffer.length &&
  crypto.timingSafeEqual(calcHmacBuffer, responseHmacBuffer)
) {    console.log("match");
} else {
    console.log("unmatch");
}
```

第5章 決済テスト

5-1 テスト実施にあたっての注意事項

テスト仕様の詳細につきましては、『導入テストガイド』を参照してください。

以下にテスト実施の際の主な注意事項を記載します。

- テスト取引を行う場合は MDK 設定ファイルのダミーモードを必ず“1”に設定してください。

```
#ダミーモード（テスト時のみ指定）
DUMMY_REQUEST    =   1
```

- テスト取引を行うには決まった設定条件（所定のカード番号、所定の決済金額など）が必要です。
  テスト時の設定条件につきましては、『導入テストガイド』を参照してください。

- 取引テスト時には、下記の事項に注意してください。
  - ✓ 取引 ID（orderId）：任意に設定してください。重複しないように設定する必要があります。また、テスト取引で使用した取引 ID を、本番取引で再度使用することはできません。
  - ✓ 金額：有効な金額データをセットしてください。（“-”（マイナス）や、小数点の入った数値は使用できません）
  - ✓ ダミーモードで実行した取引に対する本番モードでの処理要求や、本番モードで実行した取引に対するダミーモードでの処理要求は、必ずエラーになります。

5-2 MAP を利用したテスト取引の結果確認

店舗様サイトから行ったテスト取引の処理結果は、弊社が提供する管理ツール MAP にログインし、検索・参照することが可能です。

MAP のご利用方法の詳細は、MAP のヘルプ機能から『ご利用ガイド』を参照してください。

※MAP とは

Merchant Administration Portal の略で、決済の管理・運用上の各種情報・機能を提供する Web 型の管理ツールです。

URL： https://admin3.veritrans.co.jp/maps/