



# D-Mode を使用した発信メール配信アプライアンスの最適化

この章は、次の項で構成されています。

- [機能の概要：最適化された発信配信の D-Mode](#) (1 ページ)
- [最適化された発信メール配信のアプライアンスの設定](#) (3 ページ)
- [IronPort Mail Merge \(IPMM\) を使用した大量のメールの送信](#) (4 ページ)

## 機能の概要：最適化された発信配信の D-Mode

D-Mode は、特定の E メールセキュリティアプライアンスを発信電子メール配信向けに最適化する、キーでイネーブルにされる機能です。着信メール処理に特有の機能は、D-Mode ではディセーブルになっています。

- [D-Mode 対応アプライアンス固有の機能](#) (1 ページ)
- [D-Mode 対応アプライアンスでディセーブルになっている標準機能](#) (2 ページ)
- [D-Mode 対応アプライアンスに適用される標準機能](#) (2 ページ)

## D-Mode 対応アプライアンス固有の機能

- **256 の仮想ゲートウェイ アドレス：**Cisco Virtual Gateway テクノロジーを使用すると、個別の IP アドレス、ホスト名およびドメインを使用してホストするすべてのドメインのエンタープライズメールゲートウェイを設定して、同じ物理アプライアンス内でホストしながら、これらのドメインの個別の企業電子メールポリシー拡張およびアンチスパム戦略を作成できます。[電子メールを受信するためのゲートウェイの設定](#)にある「リスナーのカスタマイズ」に関する情報を参照してください。
- **IronPort Mail Merge (IPMM)：**IronPort Mail Merge (IPMM) を使用すると、個別の個人向けメッセージをカスタマーシステムから生成する手間を省くことができます。ユーザは、数千の個別メッセージを生成し、メッセージ生成システムと電子メールゲートウェイ間で送信する必要がなくなるため、システムにかかる負荷が軽減され、電子メール配信のスループットが向上します。詳細については、[IronPort Mail Merge \(IPMM\) を使用した大量のメールの送信](#) (4 ページ) を参照してください。

- リソースを節約するバウンス設定：D-Mode-enabled アプライアンスを設定して、ブロックされている可能性のある宛先を検出し、その宛先へのすべてのメッセージをバウンスできます。詳細については、[リソースを節約するバウンス設定の構成（4 ページ）](#) を参照してください。
- 発信配信のパフォーマンスの向上

## D-Mode 対応アプライアンスでディセーブルになっている標準機能

- IronPort Anti-Spam スキャンおよびオン/オフボックス スпам隔離：アンチスパム スキャンは、通常、着信メールに関係するため、IronPort Anti-Spam スキャン エンジンがディセーブルにされます。したがって、スパム対策の章は適用されません。
- アウトブレイク フィルタ：アウトブレイク フィルタは、着信メールの隔離に使用されるため、D-Mode-enabled アプライアンスではディセーブルになっています。したがって、アウトブレイク フィルタの章の情報は適用されません。
- SenderBase Network Participation 機能：SenderBase Network Participation は、着信メールに関する情報を報告するため、D-Mode-enabled アプライアンスではディセーブルになっています。したがって、SenderBase Network Participation に関する情報は適用されません。
- レポーティング：レポーティング機能は限定されます。一部のレポートは使用できません。発生するレポーティングも、パフォーマンス上の理由により、非常に限定的なレベルで実行するように設定されています。



(注) D-Mode-enabled アプライアンスの電子メール セキュリティ モニタ概要レポートに示される合計には、これらの機能が D-Mode-enabled アプライアンスでディセーブルにされている場合でも、スパム、および陽性と疑わしいスパムの数が、誤って含まれる可能性があります。

- データ損失防止：発信メッセージの DLP スキャンは、D-Mode-enabled アプライアンスでディセーブルになっています。

## D-Mode 対応アプライアンスに適用される標準機能

表 1: D-Mode-enabled アプライアンスに含まれる AsyncOS 機能

機能	詳細情報
アンチウイルス スキャン	<a href="#">アンチウイルス</a> を参照してください。
DomainKeys 署名	DKIM/DomainKeys は、送信者により使用される署名キーに基づいて電子メールの信頼性を確認する方式です。 <a href="#">電子メール認証</a> を参照してください。
集中管理	<a href="#">クラスタを使用した中央集中型管理</a> を参照してください。

機能	詳細情報
配信スロットリング	各ドメインに対して、一定期間でシステムが超えることができない、接続および受信者の最大数を割り当てることができます。「グッドネイバー」テーブルは、 <code>destconfig</code> コマンドで定義されます。  詳細については、 <a href="#">宛先制御による電子メール配信の管理</a> を参照してください。
バウンス検証	バウンス メッセージの信頼性を検証します。 <a href="#">バウンス検証</a> を参照してください。
委任管理	<a href="#">管理タスクの分散</a> を参照してください。
トレース (デバッグ)	<a href="#">テストメッセージを使用したメールフローのデバッグ：トレース</a> を参照してください。
VLAN、NIC ペアリング	<a href="#">高度なネットワーク構成</a> を参照してください。
オプションのアンチウイルス エンジン	オプションのアンチウイルス スキャンを追加することで、アウトバウンドメッセージの完全性を保証できます。 <a href="#">アンチウイルス スキャンの概要</a> を参照してください。

## 最適化された発信メール配信のアプライアンスの設定

**ステップ 1** 提供されているライセンス キーを適用します。システム セットアップ ウィザードを実行する前（アプライアンスを設定する前）に、このキーを Cisco E メール セキュリティ アプライアンスに適用する必要があります。キーの適用は、[システム管理 (System Administration)] > [ライセンスキー (Feature Key)] ページを介して、または CLI の `featurekey` コマンドを入力して行います。

(注) 前述のライセンス キーには、サンプルの Sophos または McAfee Anti-Virus の 30 日間ライセンスが含まれています。これは、アウトバウンドメールでのアンチウイルス スキャンのテストに使用できます。

**ステップ 2** アプライアンスを再起動します。

**ステップ 3** システム セットアップ ウィザード (GUI または CLI) を実行して、アプライアンスを設定します。

発信メール配信用に最適化されたアプライアンスには、アンチスパム スキャンもアウトブレイク フィルタも含まれないことに注意してください。（これらの章は無視してください）。

(注) クラスタ環境では、D-Mode 機能キーで設定されたアプライアンスを、配信パフォーマンス パッケージで設定されていない AsyncOS アプライアンスと組み合わせることはできません。

## リソースを節約するバウンス設定の構成

最適化された発信メール配信向けにアプライアンスを設定した後は、潜在的な配信問題を検出し、特定の宛先へのすべてのメッセージをバウンスするようにシステムを設定できます。



- (注) この設定を使用すると、配信不能と見なされる宛先ドメインのキューのすべてのメッセージがバウンスされます。メッセージは、配信問題が解決された後で再送信する必要があります。

### リソースを節約するバウンス設定をイネーブルにする例

```
mail3.example.com> bounceconfig

Choose the operation you want to perform:

- NEW - Create a new profile.
- EDIT - Modify a profile.
- DELETE - Remove a profile.
- SETUP - Configure global bounce settings.

[]> setup

Do you want to bounce all enqueued messages bound for a domain if the host is down? [N]>
y
```

この機能を使用する場合、最新の接続試行が 10 回連続で失敗すると、ホストは「ダウン」と見なされます。AsyncOS は、ダウン ホストを 15 分ごとにスキャンします。そのため、接続は、キューがクリアされる前に 11 回以上試行されます。

## IronPort Mail Merge (IPMM) を使用した大量のメールの送信



- (注) IronPort Mail Merge は、D-Mode-enabled アプライアンスでのみ使用可能です。

### IronPort Mail Merge の概要

IronPort Mail Merge を使用すると、個別の個人向けメッセージをカスタマー システムから生成する手間を省くことができます。ユーザは、数千の個別メッセージを生成し、メッセージ生成システムと電子メールゲートウェイ間で送信する必要がなくなるため、システムにかかる負荷が軽減され、電子メール配信のスループットが向上します。

IPMMでは、個人向けに置換されるメッセージの場所を表す変数を使用して、各メッセージの本文が作成されます。各メッセージ受信者に対して、受信電子メールアドレスおよび変数置換だけを電子メールゲートウェイに送信する必要があります。また、IPMMを使用して、受信者に応じて、送信するメッセージの本文の特定の「パーツ」を含めたり、除外したりできます（たとえば、2つの異なる国の受信者に送信するメッセージの最後に異なる著作権宣言文を含めることができます）。

## Mail Merge 機能の利点

- メール管理者にとって使いやすい。IPMMは、変数置換および一般的な多くの言語の抽象化インターフェイスを提供するため、各受信者の個人向けメッセージを簡単に作成できます。
- メッセージ生成システムの負荷を軽減する。メッセージ本文の1つのコピーと必須の置換のテーブルだけが必要であるため、ほとんどのメッセージ生成「作業」をメッセージ生成システムから、最適化された発信メール配信向けに設定されたアプライアンスに移行して、負荷を軽減できます。
- 配信スループットが改善される。数千の着信メッセージを受け取り、キューに入れるために必要なリソースを軽減することで、アプライアンスは、アウトバウンド配信パフォーマンスを大幅に改善できます。
- キューストレージの効率性が向上する。各メッセージ受信に保存する情報を減らすことで、ユーザは、D-Mode-enabled アプライアンスのキューストレージの使用効率を大幅に向上できます。

## Mail Merge の使用

### SMTP インジェクション

IPMMは、SMTPをトランスポートプロトコルとして拡張します。アプライアンスで行う特別な設定は必要ありません（デフォルトでは、IPMMは、プライベートリスナーでイネーブルにして、D-Mode-enabled アプライアンスのパブリックリスナーでディセーブルにできます）。ただし、現在、SMTPをインジェクションプロトコルとして使用していない場合は、D-Mode-enabled アプライアンスインターフェイスを介してSMTPを利用する新しいプライベートリスナーを作成する必要があります。

listenerconfig の setipmm サブコマンドを使用して、リスナーでIPMMを有効にします。詳細については、[電子メールを受信するためのゲートウェイの設定](#)を参照してください。

IPMMは、MAIL FROM と DATA の2つのコマンドを変更し、XDFNを追加することで、SMTPを変更します。MAIL FROM コマンドはXMRG FROMに、DATA コマンドはXPRTに置き換えられています。

Mail Mergeメッセージを生成するには、メッセージの生成に使用されるコマンドを特定の順序で発行する必要があります。

1. 送信ホストを示す、初期 EHLO ステートメント。
2. 各メッセージは、送信者アドレスを示す、XMRG FROM: ステートメントで始まります。
3. 各受信者は、次のように定義されます。
4. 1つ以上のXDFN変数割り当てステートメントが含まれます。これには、パーツ定義（XDFN \*PART=1,2,3...）やその他の任意の受信者固有の変数が含まれます。
5. 受信者電子メールアドレスは、RCPT TO: ステートメントで定義されます。RCPT TO: の前にあり、前述の XMRG FROM または RCPT TO コマンドの後にある任意の変数割り当ては、この受信者電子メールアドレスにマッピングされます。
6. 各パーツは、XPRT n コマンドを使用して定義されます。各パーツは、DATA コマンドと同様にピリオド (.) 文字で終了します。最後のパーツは、XPRT n LAST コマンドで定義されます。

## 変数置換

メッセージヘッダーなど、メッセージ本文の任意のパーツに、置換用の変数を含めることができます。変数は、HTML メッセージにも表示できます。変数は、ユーザが定義し、アンバサンド (&) 文字で始まり、セミコロン (;) 文字で終了する必要があります。アスタリスク (\*) で始まる変数名は、予約されているため使用できません。

## 予約変数

IPMM には、事前に定義されている 5 つの特殊な「予約」変数が含まれます。

表 2: IPMM : 予約変数

*FROM	予約変数 *FROM は、「Envelope From」パラメータから派生します。「Envelope From」パラメータは、「XMRG FROM:」コマンドにより設定されます。
*TO	予約変数 *TO は、「RCPT TO:」コマンドで設定される、エンベロープ受信者値から派生します。
*PARTS	予約変数 *PARTS は、パーツのカンマ区切りリストを含みます。これは、「RCPT TO:」で受信者を定義する前に設定され、特定のユーザが受信する「XPRT n」メッセージ本文ブロックを決定します。
*DATE	予約変数 *DATE は、現在の日付スタンプに置き換えられます。
*DK	予約変数 *DK は、DomainKeys 署名プロファイルの指定に使用されます（このプロファイルはすでに AsyncOS に存在している必要があります）。DomainKeys 署名プロファイルの作成の詳細については、 <a href="#">電子メール認証</a> を参照してください。

## メッセージの例 1

次の例のメッセージ本文（ヘッダーを含む）には、最後のメッセージで置換される、4 つの異なる変数と 5 つの置換用の場所が含まれます。同じ変数がメッセージ本文で複数回使用される

ことがあるため注意してください。また、予約変数 `&*TO;` が使用されます。これは、受信者の電子メールアドレスに置換されます。この予約変数は、個別の変数として渡す必要はありません。次の例の変数は太字で示されています。

```
From: Mr.Spacely <spacely@example.com>

To: &first_name;&last_name;&*TO;

Subject: Thanks for Being an Example.Com Customer

Dear &first_name;,
Thank you for purchasing a &color; sprocket.
```

このメッセージは、アプライアンスに一度だけインジェクトする必要があります。各受信者に対して、次の追加情報が必要です。

- 受信者の電子メール アドレス
- 変数置換の名前と値のペア

## パーツ アセンブリ

SMTP は、各メッセージ本文に単一の DATA コマンドを使用し、IPMM は、1 つ以上の XPRT コマンドを使用してメッセージを作成します。パーツは、受信者ごとに指定される順序に従ってアセンブルされます。各受信者は、任意またはすべてのメッセージパーツを受信できます。パーツは、任意の順序でアセンブルできます。

特殊な変数 `*PARTS` は、パーツのカンマ区切りリストを含みます。

たとえば、次の例のメッセージでは、2 つのパーツが含まれます。

最初のパーツには、メッセージヘッダーとメッセージ本文の一部が含まれます。2 番めのパーツには、特別なカスタマー向けに含めることができる割引価格が含まれます。

### メッセージの例 2 (パート 1)

```
From: Mr. Spacely <spacely@example.com>

To: &first_name; &last_name; &*TO;

Subject: Thanks for Being an Example.Com Customer

Dear &first_name;,

Thank you for purchasing a &color; sprocket.
```

### メッセージの例 2 (パート 2)

Please accept our offer for 10% off your next sprocket purchase.

メッセージ部分は、アプライアンスに一度だけインジェクトする必要があります。この場合、各受信者に、次の追加情報が必要です。

- 最後のメッセージに含まれる、パーツの順序付きリスト

- 受信者の電子メール アドレス
- 変数置換の名前と値のペア

## IPMM および DomainKeys 署名

IPMM は、DomainKeys 署名をサポートします。DomainKeys プロファイルを指定するには、\*DK 予約変数を使用します。次に例を示します。

```
XDFN first_name="Jane" last_name="User" color="red" *PARTS=1,2 *DK=mass_mailing_1
```

この例では、「mail\_mailing\_1」は、前に設定した DomainKeys プロファイルの名前です。

## コマンドの説明

クライアントは、IPMM メッセージをリスナーにインジェクトするときに、次のキーコマンドで拡張 SMTP を使用します。

### XMRG FROM

構文：

```
XMRG FROM: <sender email address>
```

このコマンドは、SMTP MAIL FROM: コマンドの代わりに使用されます。これは、次に IPMM メッセージがあることを示します。IPMM ジョブは、XMRG FROM: コマンドで開始されます。

### XDFN

構文：

```
XDFN <KEY=VALUE> [KEY=VALUE]
```

XDFN コマンドは、受信者別のメタデータを設定します。キーと値のペアは、オプションでかぎカッコまたは角カッコで囲むことができます。

\*PARTS は、XPRT コマンド（以下を参照）で定義されているように、インデックス番号を示す特殊な予約変数です。\*PARTS 変数は、整数のカンマ区切りリストとして分割されます。整数は、XPRT コマンドにより定義されているように送信される本文パーツと一致します。その他の予約変数には、\*FROM、\*TO および \*DATE があります。

### XPRT

構文：

```
XPRT index_number LAST
```

Message

.

XPRT コマンドは、SMTP DATA コマンドの代わりに使用されます。このコマンドは、コマンド入力後にメッセージパーツの送信者を受け取ります。コマンドは、行の末尾に単一のピリオドを付けて完了します（これは、SMTP DATA コマンドを完了する方法と同じです）。



特殊キーワード **LAST** は、Mail Merge ジョブの最後を示します。これは、インジェクトされる最後のパーツを指定するときに使用する必要があります。

LAST キーワードが使用されると、メッセージがキューに入り、配信が始まります。

## 変数定義に関する注意事項

- XDFN コマンドで変数を定義する場合、実際のコマンドラインは、システムの物理的制限を超えることはできないため注意してください。D-Mode-enabled アプライアンスの場合、この制限は、1 行あたり 4 KB です。ホストシステムによっては、しきい値がこれより低くなる場合があります。非常に長いコマンドラインで複数の変数を定義する場合は注意してください。
- 変数キーと値のペアを定義する場合、スラッシュ「/」文字を使用して、特殊文字をエスケープできます。これは、メッセージ本文に、誤って変数定義と置換される可能性がある HTML 文字エンティティが含まれる場合に役に立ちます（たとえば、文字エンティティ™ は、商標文字の HTML 文字エンティティを定義します）。コマンド XDFN trade=foo を作成して、HTML 文字エンティティ「™」を含む IPMM メッセージを作成した場合、アセンブルされるメッセージには、商標文字ではなく、変数置換（「foo」）が含まれます。これは、GET コマンドを含む URL で使用されることがあるアンパサンド文字「&」の場合も同じです。

## IPMM カンバセーションの例

次に、メッセージの例 2（前述の例）での IPMM カンバセーションの例を示します。このメッセージは、この例の 2 人の受信者「Jane User」および「Joe User」に送信されます。

この例では、太字フォントは、D-Mode-enabled アプライアンスとの手動による SMTP カンバセーションで入力する内容です。また、モノスペース タイプのフォントは、SMTP サーバからの応答を表し、イタリック体フォントは、コメントまたは変数を表します。

接続が確立されます。

**220 ESMTP**

**EHLO foo**

**250** - ehlo responses from the listener enabled for IPMM

カンバセーションが開始されます。

**XMRG FROM:**<user@domain.com> [Note: This replaces the **MAIL FROM:** SMTP command.]

**250 OK**

変数およびパーツが各受信者に設定されます。

**XDFN first\_name="Jane" last\_name="User" color="red" \*PARTS=1,2**

*[Note: This line defines three variables (first\_name, last\_name, and color) and then*

*uses the \*PARTS reserved variable to define that the next recipient defined will receive message parts numbers 1 and 2.]*

250 OK

RCPT TO:<jane@company.com>

250 recipient <jane@company.com> ok

XDFN first\_name="Joe" last\_name="User" color="black" \*PARTS=1

*[Note: This line defines three variables (first\_name, last\_name, and color) and then uses the \*PARTS reserved variable to define that the next recipient defined will receive message parts numbers 1 only.]*

RCPT TO:<joe@company1.com>

250 recipient <joe@company1.com> ok

次に、パーツ 1 が送信されます。

XPRT 1 *[Note: This replaces the DATA SMTP command.]*

354 OK, send part

From: Mr. Spacely <spacely@example.com>

To: &first\_name; &last\_name; &\*TO;

Subject: Thanks for Being an Example.Com Customer

&\*DATE;

Dear &first\_name;;

Thank you for purchasing a &color; sprocket.

.

次に、パーツ 2 が送信されます。LAST キーワードは、パーツ 2 がアセンブルする最後のパーツであることを示すときに使用されます。

XPRT 2 LAST

Please accept our offer for 10% off your next sprocket purchase.

.

250 Ok, mailmerge message enqueued

「250 Ok, mailmerge message queued」は、メッセージが受け取られたことを示します。

この例に基づいて、受信者 Jane User は、このメッセージを受信します。

From: Mr. Spacely <spacely@example.com>

To: Jane User <jane@company.com>

Subject: Thanks for Being an Example.Com Customer

*message date*

Dear Jane,

Thank you for purchasing a red sprocket.

Please accept our offer for 10% off your next sprocket purchase.

受信者 Joe User は、このメッセージを受信します。

From: Mr. Spacely <spacely@example.com>

To: Joe User <joe@company1.com>

Subject: Thanks for Being an Example.Com Customer

*message date*

Dear Joe,

Thank you for purchasing a black sprocket.

## コード例

シスコは、一般的なプログラミング言語でライブラリを作成して、IPMM メッセージを IPMM 対応のアプライアンスリスナーにインジェクトするタスクを抽象化します。IPMM ライブラリ の使用例については、シスコ カスタマー サポートにお問い合わせください。コードは、構文 説明のために広範囲にわたってコメント化されています。

