



DataSpider BPM 2.5

# クラスタ構成ガイド

(第一版)



## 目次

1.	はじめに.....	2
1.1.	本ドキュメントについて.....	2
1.2.	注意事項.....	3
1.2.1.	お客さまへのお願い.....	3
1.2.2.	商標について.....	3
1.3.	マークについて.....	4
2.	データについて.....	5
2.1.	データベース.....	5
2.2.	動的ファイル.....	5
3.	クラスタ構成.....	7
3.1.	DataSpider BPM の待機に関する注意事項.....	7
3.2.	クラスタ構成図.....	8
3.3.	クラスタの構築手順.....	10
3.3.1.	インストール.....	10
3.3.2.	設定情報の同期.....	11
3.3.3.	動的ファイルの同期設定.....	11
3.3.4.	PostgreSQL データベースの同期設定.....	12
3.4.	フェイルオーバーの手順.....	15
3.5.	モジュールの同期.....	16
4.	稼働状況の監視方法.....	17
4.1.	DataSpider BPM アプリケーションの死活監視.....	17
5.	障害復旧時の動作.....	18
5.1.	イベントの動作について.....	18
5.2.	締め切り通知の動作について.....	19

## 1. はじめに

### 1.1. 本ドキュメントについて

本ドキュメントでは、DataSpider BPM(製品版/評価版 共通)において、DataSpider BPM システムのクラス  
タ構成について説明しています。

本ドキュメントの内容は、バージョン 2.5 に基づきます。

## 1.2. 注意事項

### 1.2.1. お客さまへのお願い




- 本ソフトウェアの著作権は株式会社アプレッソまたはそのライセンサーが所有しています。
- 本ソフトウェアおよび本ドキュメントを無断で複製、転載することを禁止します。
- 本ドキュメントは万全を期して作成されていますが、万一不明な点や誤り、記載もれなど、お気づきの点がございましたら弊社までご連絡ください。
- 本ソフトウェアは使用者の責任でご使用ください。ご使用の結果、万トラブルおよび訴訟等が発生しましても、あらゆる直接、または間接の損害および損失につきまして、弊社は一切責任を負わないものとしします。あらかじめご了承ください。
- 本ソフトウェアの仕様や本ドキュメントに記載されている内容は、改善のため予告なしに変更されることがあります。
- 本ソフトウェアの使用には、ソフトウェアライセンス契約が必要で、株式会社アプレッソまたはそのライセンサーの重要な業務機密と独自の情報が含まれており、日本国政府の著作権法で保護されています。株式会社アプレッソまたはそのライセンサーのソフトウェアと本ドキュメントの無断使用は、損害賠償、刑事訴訟の対象となります。

### 1.2.2. 商標について

- APPRESSO、APPRESSO ロゴ、アプレッソ、DataSpider、DataSpider マーク、データスパイダー、Servista、Servista ロゴ、サービススタは、株式会社アプレッソの商標または登録商標です。
- APPRESSO、APPRESSO ロゴ、アプレッソ、DataSpider、DataSpider マーク、データスパイダー、Servista、Servista ロゴ、サービススタ以外の会社名、製品名、サービス名等は、各社の登録商標または商標です。
- 個々のページに表示・記載されたこれら商標等の複製・転用を禁止致します。

## 1.3. マークについて

本ドキュメント内で使用しているマークについての説明は以下の通りです。

マーク	説明
	操作や設定に関するヒントであることを表します。
	操作や設定に関する注意事項や制限事項であることを表します。
	詳細な説明が別の項目に記載されていることを表します。

また、説明は次の規則に沿って行われています。

- 画面に表示されるメニュー名、タブ名、プロパティ項目名および値、ボタン名は[]で囲んで表します。また、それ以外の機能名や画面のタイトル、名称のないものは「」で囲んで前者と区別しています。
- \$DSBPM\_HOME は、DataSpider BPM をインストールしたディレクトリを表します。デフォルトでは、「C:¥DSBPM」となります。
- \$DSS\_HOME は、DataSpider Servista (Server)のインストールディレクトリを表します。
- \$JRE\_HOME は、DataSpider BPM のインストール時に選択した JavaVM(JRE)のインストールディレクトリを表します。
  - JRE をインストールした場合： 例) C:¥Program Files¥Java¥jre1.8.0\_141
  - JDK をインストールした場合： 例) C:¥Program Files¥Java¥jdk1.8.0\_141¥jre
- \$PSQL\_HOME は、PostgreSQL データベースのインストールディレクトリを表します。
- <>で囲まれた名称は、可変であることを表します。  
例: http://<ホスト名または IP アドレス>:18080/userweb/Login\_show

## 2. データについて

DataSpider BPM の運用時に、読み込みおよび書き込みを行うデータにはどのようなものがあるのかを把握し、クラスタ構成を構築する上で注意すべき内容を説明します。

### 2.1. データベース

作成済みのデータ(アプリ、ユーザ、組織などすべての作成済みデータ)および実行履歴データ(プロセス、タスクなどすべての実行履歴データ)は、PostgreSQL データベースの中に保持しています。

### 2.2. 動的ファイル

DataSpider BPM の運用時に、データベース以外に格納される読み込みおよび書き込みを行うファイルデータは以下の通りです。

- プロセスやオープンチャットで使用するファイル

#### 【格納場所】

`$DSBPM_HOME¥file` ディレクトリ直下

#### 【説明】

プロセス実行中のタスクなどで添付したファイルデータ(ファイル型のデータ項目に保持しているデータ)は、`$DSBPM_HOME¥file` ディレクトリ直下に保持しています。プロセスの処理中にファイルを添付した際には、このディレクトリに添付ファイルをアップロードし、添付ファイルをダウンロードした際には、このディレクトリからダウンロードされます。

また、オープンチャットの投稿時に添付したファイルも同様に、`$DSBPM_HOME¥file` ディレクトリ直下に保持しています。これらの格納されたファイルは、添付したファイル名ではなく、システム管理番号が付けられた以下のファイル名で DataSpider BPM が保持管理します。

- ファイル型のデータ項目に添付した場合：<ファイル管理 ID>.拡張子  
例) 1234.xls
- オープンチャットの投稿時に添付した場合：f\_<ファイル管理 ID>.拡張子  
例) f\_4567.ppt



システムが内部的に管理しているファイルですので、直接変更や削除は行わないでください。



ファイルデータは、プロセス処理中に限らず終了後も永続的に保持され、ファイルをダウンロードするには必ず存在する必要があります。よって、各ノードに DataSpider BPM をインストールするクラスタ構成の場合は、パッシブノードに必ずファイル同期を行う必要があります。

- オープンチャットが使用するインデックスファイル

**【格納場所】**

`$DSBPM_HOME¥index` ディレクトリ直下

**【説明】**

オープンチャットで投稿した内容はデータベースに保持されますが、DataSpider BPM では投稿内容の参照および検索で使用するためのインデックスファイルを自動生成または更新します。

- DataSpiderBPM の起動時：インデックスファイルが存在しない場合は、新規に生成します。
- オープンチャットで投稿時：インデックスファイルを更新します。



通知投稿やタスク処理時に DataSpider BPM が自動投稿する場合も同様に更新します。



各ノードに DataSpider BPM をインストールするクラスタ構成の場合などでは、パッシブノードにファイル同期を行うことをおすすめしますが、フェイルオーバー時に初めて起動するようなワールドスタンバイ構成の場合は、起動時に新たに自動生成されるためファイル同期は必須ではありません。ただし、インデックスファイルを同期しない場合は、起動前に`$DSBPM_HOME¥index` ディレクトリ直下にはディレクトリおよびファイルが何も存在しないことを確認してください。

## 3. クラスタ構成

DataSpider BPM のクラスタ構成は、アクティブノードとパッシブノードにおいて、DataSpider BPM と PostgreSQL データベースの両方を冗長化した構成をサポートしています。アクティブノードは本番系とも呼ばれ、パッシブノードは待機系とも呼ばれます。

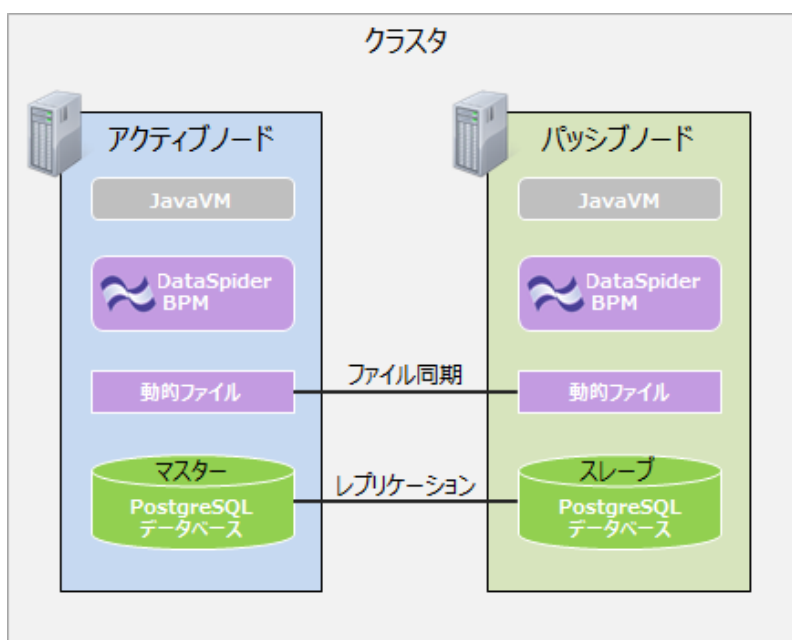
### 3.1. DataSpider BPM の待機に関する注意事項

DataSpider BPM は、複数ノード間で同時使用または起動状態(ホットスタンバイ)が行えません。クラスタ環境を構築する場合は、パッシブノードの DataSpider BPM は停止状態(コールドスタンバイ)にしてください。



## 3.2. クラスタ構成図

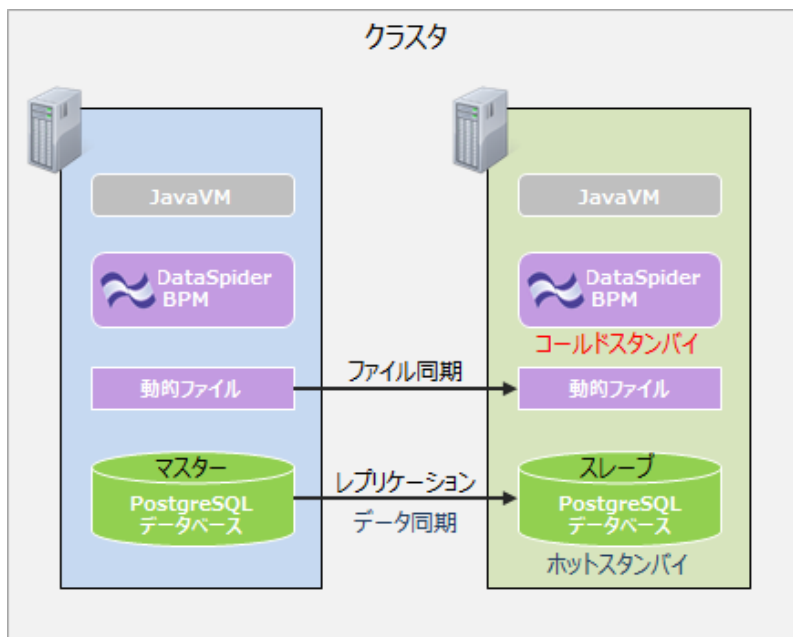
JavaVM(JRE)と DataSpider BPM および PostgreSQL データベースはアクティブノードとパッシブノードそれぞれにインストールします。動的ファイルはパッシブノードにフェイルオーバー時にコピーするか、ファイル同期ソフトなどを使用してリアルタイムにファイル同期します。データベースのデータは PostgreSQL のレプリケーション機能を使用し、パッシブノードにリアルタイムでデータ同期します。



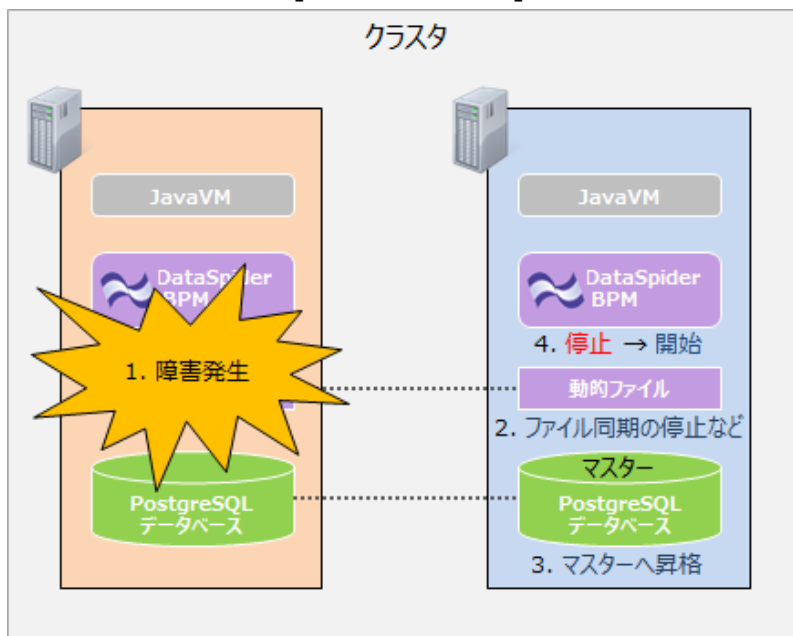
システム構成に関して、以下の注意事項があります。

- 各ノードのシステム時刻は合わせる必要があります。
- 各ノードともにインストールする JavaVM(JRE)のバージョンは合わせる必要があります。
- 各ノードともにインストールする DataSpider BPM のバージョンは合わせる必要があります。
- 各ノードともにインストールする PostgreSQL データベースのバージョンは合わせる必要があります。
- パッシブノードの DataSpider BPM は、停止状態(コールドスタンバイ)で待機します。
- パッシブノードの PostgreSQL データベースは、アクティブノードのデータベースのスレーブとしてレプリケーション設定し、起動状態(ホットスタンバイ)で待機します。
- フェイルオーバー時には、パッシブノードの PostgreSQL データベースをスレーブからマスターに昇格させたのち DataSpider BPM を起動します。

[正常稼働時]



[フェイルオーバー時]



### 3.3. クラスタの構築手順

クラスタの構築は以下のような流れになります。

1. インストール
2. モジュールの同期
3. 設定情報の同期
4. 動的ファイルの同期設定
5. PostgreSQL データベースの同期設定

#### 3.3.1. インストール



インストール方法に関する詳細は、インストールガイドを参照してください。

##### [アクティブノード]

1. JavaVM(JRE)をインストールします。
2. PostgreSQL データベースをインストールし、基本設定を行います。
3. DataSpider BPM をインストールし、基本設定を行います。



インストール時には、Windows サービスとして登録してください。

##### [パッシブノード]



インストールするバージョンは、アクティブノードと合わせてください。

1. JavaVM(JRE)をインストールします。
2. PostgreSQL データベースをインストールし、基本設定を行います。
3. DataSpider BPM をインストールし、基本設定を行います。



インストール時には、Windows サービスとして登録してください。



上記のほかに、動的ファイルをリアルタイムに同期する場合は、ファイル同期ソフトをインストールし、適宜設定してください。

### 3.3.2. 設定情報の同期

パッシブノードにて、アクティブノードで設定した設定情報の同期を行う必要があります。

同期方法は、アクティブノードで変更を行った内容をパッシブノードの対象ファイルに記載・設定する、またはファイルのコピーを行います。



変更やコピーを行う前には、バックアップを取ることをおすすめします。

- DataSpider BPM 設定ファイル
  - `$DSBPM_HOME¥qbpms.config`

- SSL 通信設定ファイル



DataSpider BPM に SSL 通信設定を行っている場合が対象となります。

- `$DSBPM_HOME¥apache-tomcat¥conf¥.keystore`
- `$DSBPM_HOME¥conf¥server.xml`
- `$JRE_HOME¥lib¥security¥cacerts`

- シングルサインオン設定ファイル



DataSpider BPM にシングルサインオンの設定を行っている場合が対象となります。

- `$DSBPM_HOME¥qbpms.config` の `qbpms.saml.keystore.file` に記載されているファイル

### 3.3.3. 動的ファイルの同期設定

パッシブノードにて、アクティブノードに保持している動的ファイルの同期設定を行う必要があります。

同期方法は、以下のディレクトリ直下のファイルをファイル同期ソフトなどを使用し、パッシブノードにリアルタイムでファイルを同期します。



ファイル同期ソフトなどを使用する場合のみ必要な作業となります。



データがすでに存在している場合は、作業を行う前にはバックアップを取ることをおすすめします。

- (必須)プロセスやオープンチャットで使用するファイル：`$DSBPM_HOME¥file` ディレクトリ
- (任意)オープンチャットが使用するインデックスファイル：`$DSBPM_HOME¥index` ディレクトリ

### 3.3.4. PostgreSQL データベースの同期設定

パッシブノードにて、アクティブノードに保持している PostgreSQL データベースファイルの同期設定を行う必要があります。

同期方法は、PostgreSQL データベースのレプリケーション機能を使用し、パッシブノードにリアルタイムでデータを同期します。同期設定の手順は以下を参照してください。



\$PSQL\_HOME\data は、インストール時に指定した PostgreSQL データベースのデータディレクトリを表します。デフォルトでは、「C:\Program Files\PostgreSQL\9.5\data」となります。



PostgreSQL データベースの同期設定の際には、事前に以下のサービスは停止してください。

- アクティブノードおよびパッシブノードの DataSpider BPM
- アクティブノードおよびパッシブノードの PostgreSQL データベース

#### ● アクティブノード：マスター側の設定

1. データディレクトリに配置されている設定ファイル(postgresql.conf)に以下の設定を行います。

- \$PSQL\_HOME\data\postgresql.conf

```
listen_addresses = '*'
wal_level = hot_standby
max_wal_senders = 2           ※スレーブの最大数を適宜設定します。
wal_keep_segments = 8        ※WALログの数を適宜設定します。
synchronous_standby_names = 'slave'  ※レプリケーションするスレーブ名を適宜設定します。
```

2. データディレクトリに配置されている設定ファイル(pg\_hba.conf)に以下の設定を行います。

- \$PSQL\_HOME\data\pg\_hba.conf

```
host      replication      postgres      <スレーブ側の IP アドレス>/32      trust
```

3. Windows サービスから PostgreSQL データベースのサービスを起動します。

#### ● パッシブノード：スレーブ側の設定

1. データディレクトリ内にあるファイルおよびフォルダを安全な場所にコピーしてバックアップを取ったのち、すべて削除します(次の手順の pg\_basebackup コマンドを使用する際にデータディレクトリ内にデータが残っているとエラーとなるため)。

- \$PSQL\_HOME\data ディレクトリ直下のファイルおよびフォルダ

2. コマンドプロンプトを開き、`$PSQL_HOME¥bin` に移動して以下のコマンドを実行します。

```
pg_basebackup -h <マスター側の IP アドレス> -p <マスター側のポート番号> -U
postgres -D <スレーブ型のデータディレクトリ> --xlog --progress
```

```
C:¥Program Files¥PostgreSQL¥9.5¥bin> pg_basebackup -h "192.168.50.28" -p 5432 -U
postgres -D "C:¥Program Files¥PostgreSQL¥9.5¥data" --xlog --progress [RETURN]
```



`pg_basebackup` コマンドは管理者として実行する必要があります。

3. データディレクトリに配置されている設定ファイル(`postgresql.conf`)に以下の設定を行います

➤ `$PSQL_HOME¥data¥postgresql.conf`

```
#コメントアウトされていることを確認
#synchronous_standby_names = "

hot_standby = on
```

4. `$PSQL_HOME¥share` に配置されている設定ファイル(`recovery.conf.sample`)をコピーし、データディレクトリに配置します。

➤ `$PSQL_HOME¥share¥recovery.conf.sample` コピー元

➤ `$PSQL_HOME¥data¥recovery.conf.sample` コピー先


5. コピーした `recovery.conf.sample` を `recovery.conf` にリネームし、以下の設定を行います。

➤ `$PSQL_HOME¥data¥recovery.conf`

```
standby_mode = on
primary_conninfo = 'host=<マスター側の IP アドレス> port=<マスター側のポート番号>
user=postgres password=<マスター側のパスワード> application_name=slave'
```

6. Windows サービスから PostgreSQL データベースのサービスを起動します。

- 同期設定(レプリケーション)の動作確認
  1. マスター側の PostgreSQL データベースにて、任意のデータベースを作成します。
  2. スレーブ側の PostgreSQL データベースにて、マスター側で作成したデータベースが同期されていることを確認します。
  
- アクティブノードの DataSpider BPM の起動
  1. アクティブノードの DataSpider BPM を起動します。

 パッシブノードの DataSpider BPM は、正常稼働時はつねに停止状態(コールドスタンバイ)となっている必要があります。

### 3.4. フェイルオーバーの手順

アクティブノードで障害が発生した場合のフェイルオーバーの手順は以下の通りです。

1. アクティブノードの DataSpider BPM が起動していないことを確認します。起動している場合は、DataSpider BPM を停止させてください。
2. アクティブノードのマスター側の PostgreSQL データベースを停止させます。コマンドプロンプトを開き、\$PSQL\_HOME¥bin に移動して以下のコマンドを実行します。

```
pg_ctl -D <マスター側のデータディレクトリ> stop -m fast
```

```
C:¥Program Files¥PostgreSQL¥9.5¥bin> pg_ctl -D "C:¥Program Files¥PostgreSQL ¥9.5¥data"
stop -m fast [RETURN]
```

 pg\_ctl コマンドは管理者として実行する必要があります。

3. 動的ファイルについて、以下の操作を行います。
  - ファイル同期ソフトなどを使用し、リアルタイムに同期している場合
    - アクティブノードからパッシブノードへの動的ファイルの同期処理を停止します。
  - ファイル同期ソフトなどを使用していない場合
    - アクティブノードの以下のファイルをパッシブノードの同ディレクトリへコピーします。
      - (必須) \$DSBPM\_HOME¥file ディレクトリ直下のファイルすべて
      - (任意) \$DSBPM\_HOME¥index ディレクトリ直下のディレクトリおよびファイルすべて
4. パッシブノードの PostgreSQL データベースをスレーブからマスターに昇格させます。コマンドプロンプトを開き、\$PSQL\_HOME¥bin に移動して以下のコマンドを実行します。

```
pg_ctl -D <スレーブ側のデータディレクトリ> promote
```

```
C:¥Program Files¥PostgreSQL¥9.5¥bin> pg_ctl -D "C:¥Program Files¥PostgreSQL ¥9.5¥data"
promote [RETURN]
```

 pg\_ctl コマンドは管理者として実行する必要があります。

5. パッシブノードの DataSpider BPM を起動します。



### 3.5. モジュールの同期

アクティブノードに DataSpider BPM の修正パッチモジュールの適用を行った場合には、適用したモジュールをパッシブノードにもインストールし、モジュールの同期を行う必要があります。



修正パッチモジュールの適用方法は、モジュールに含まれている README を参照してください。

## 4. 稼働状況の監視方法

ここでは、DataSpider BPM アプリケーションの死活監視方法のみを説明しています。PostgreSQL データベースの監視や障害箇所を特定したい場合は、システム運用監視ガイドを参照してください。

### 4.1. DataSpider BPM アプリケーションの死活監視

DataSpider BPM の稼働状況を監視する方法を説明します。

#### [監視方法]

以下の HTTP リクエストを発行します。

`http(s)://<ホスト名または IP アドレス>:<ポート番号>/userweb/Login_show`

例) `http://bpmserver:18080/userweb/Login_show`

#### [監視結果の判定]

- レスポンスコードが正常終了「200」場合は、DataSpider BPM が稼働状態です。

## 5. 障害復旧時の動作

障害復旧時に、イベントおよび締め切り通知の機能がどのように動作するかを説明します。

### 5.1. イベントの動作について

- タイマー開始イベント
  - 開始時刻前に障害が発生し、開始時刻[前]に復旧した場合は、タイマー開始イベントは発生しません。
  - 開始時刻前に障害が発生し、開始時刻[後]に復旧した場合は、タイマー開始イベントは発生しません。
- タイマー中間イベント
  - イベント到達後の再開時刻待ちの状態障害が発生し、再開時刻[前]に復旧した場合は、再開時刻でタイマー中間イベントが発生します。
  - イベント到達後の再開時刻待ちの状態障害が発生し、再開時刻[後]に復旧した場合は、復旧した時点でタイマー中間イベントが発生します。
- タイマー境界イベント（[タスクを異常終了させる]を設定している場合）
  - タスク到達後の締め切り時刻待ちの状態障害が発生し、締め切り時刻[前]に復旧した場合は、締め切り時刻が来た時点でタイマー境界イベントが発生します。
  - タスク到達後の締め切り時刻待ちの状態障害が発生し、締め切り時刻[後]に復旧した場合は、復旧した時点でタイマー境界イベントが発生します。
- メッセージ受信中間イベント
  - イベント到達後のメッセージ受信待ちの状態障害が発生し、復旧[前]に外部システムからメッセージを受信しても、メッセージを受信することができません。
  - イベント到達後のメッセージ受信待ちの状態障害が発生し、復旧[後]に外部システムからメッセージを受信すると、メッセージを受信することができます。
- メッセージ送信中間イベント(DataSpider Servista)
  - イベント到達後の DataSpider Servista 側スクリプト処理完了待ちの状態障害が発生し、復旧した場合は、DataSpider Servista 側スクリプトの処理完了前後に関わらずイベント処理は「処理失敗」となり、復旧した時点で次にトークンが進みます。



DataSpider Servista からの受信データを受け取ることはできません。

## 5.2. 締め切り通知の動作について

- 締め切り通知: メール
  - 締め切り通知設定のあるヒューマンタスク到達後の通知時刻前に障害が発生し、通知時刻[前]に復旧した場合は、通知時刻にメール送信します。
  - 締め切り通知設定のあるヒューマンタスク到達後の通知時刻前に障害が発生し、通知時刻[後]に復旧した場合は、復旧した時点でメール送信します。
  
- 締め切り通知: お知らせ(通知メッセージ)
  - 締め切り通知設定のあるヒューマンタスク到達後の通知時刻前に障害が発生し、通知時刻[前]に復旧した場合は、通知時刻にメッセージ投稿します。
  - 締め切り通知設定のあるヒューマンタスク到達後の通知時刻前に障害が発生し、通知時刻[後]に復旧した場合は、復旧した時点でメッセージ投稿します。

## **DataSpider BPM 2.5 クラスタ構成ガイド (第一版)**

---

最終更新日 2017 年 10 月 30 日

株式会社アプレッソ

〒107-0052 東京都港区赤坂 1-8-1 赤坂インターシティ AIR 19F

電話：03-4321-1111

---