

# Veritas NetBackup 並列スト リームフレームワーク SDK 開発者ガイド

UNIX および Linux の場合

# Veritas 並列ストリームフレームワーク SDK 開発者ガイド

最終更新日: 2018-10-18

マニュアルバージョン: NetBackup 8.1.2

## 法的通知と登録商標

Copyright © 2018 Veritas Technologies LLC. All rights reserved.

Veritas、Veritas ロゴ、NetBackup は Veritas Technologies LLC または同社の米国とその他の国における関連会社の商標または登録商標です。その他の会社名、製品名は各社の登録商標または商標です。

この製品には、サードパーティの所有物であることをベリタスが示す必要のあるサードパーティソフトウェア（「サードパーティプログラム」）が含まれている場合があります。サードパーティプログラムの一部は、オープンソースまたはフリーソフトウェアライセンスで提供されます。本ソフトウェアに含まれる本使用許諾契約は、オープンソースまたはフリーソフトウェアライセンスでお客様が有する権利または義務を変更しないものとします。このベリタス製品に付属するサードパーティの法的通知文書は次の場所から入手できます。

<https://www.veritas.com/about/legal/license-agreements>

本書に記載されている製品は、その使用、コピー、頒布、逆コンパイルおよびリバースエンジニアリングを制限するライセンスに基づいて頒布されます。Veritas Technologies LLC からの書面による許可なく本書を複製することはできません。

本書は、現状のままで提供されるものであり、その商品性、特定目的への適合性、または不侵害の暗黙的な保証を含む、明示的あるいは暗黙的な条件、表明、および保証はすべて免責されるものとします。ただし、これらの免責が法的に無効であるとされる場合を除きます。Veritas Technologies LLC は、本書の提供、内容の実施、また本書の利用によって偶発的あるいは必然的に生じる損害については責任を負わないものとします。本書に記載の情報は、予告なく変更される場合があります。

ライセンス対象ソフトウェアおよび資料は、FAR 12.212 の規定によって商業用コンピュータソフトウェアと見なされ、場合に応じて、FAR 52.227-19「Commercial Computer Software - Restricted Rights」、DFARS 227.7202、「Commercial Computer Software and Commercial Computer Software Documentation」、その後継規制の規定により制限された権利の対象となります。業務用またはホスト対象サービスとしてベリタスによって提供されている場合でも同様です。米国政府によるライセンス対象ソフトウェアおよび資料の使用、修正、複製のリリース、実演、表示または開示は、本使用許諾契約の条項に従ってのみ行われるものとします。

Veritas Technologies LLC  
500 E Middlefield Road  
Mountain View, CA 94043

<http://www.veritas.com>

## テクニカルサポート

テクニカルサポートは世界中にサポートセンターを設けています。すべてのサポートサービスは、お客様のサポート契約およびその時点でのエンタープライズテクニカルサポートポリシーに従って提供されます。サポートサービスとテクニカルサポートへの問い合わせ方法については、次の弊社の **Web** サイトにアクセスしてください。

[https://www.veritas.com/support/ja\\_JP.html](https://www.veritas.com/support/ja_JP.html)

次の URL で Veritas Account の情報を管理できます。

<https://my.veritas.com>

既存のサポート契約に関する質問については、次に示す地域のサポート契約管理チームに電子メールでお問い合わせください。

世界全域 (日本を除く)

[CustomerCare@veritas.com](mailto:CustomerCare@veritas.com)

Japan (日本)

[CustomerCare\\_Japan@veritas.com](mailto:CustomerCare_Japan@veritas.com)

## マニュアル

マニュアルの最新バージョンがあることを確認してください。各マニュアルには、2 ページに最終更新日付が記載されています。最新のマニュアルは、次のベリタス **Web** サイトで入手できます。

<https://sort.veritas.com/documents>

## マニュアルに対するご意見

お客様のご意見は弊社の財産です。改善点のご指摘やマニュアルの誤謬脱漏などの報告をお願いします。その際には、マニュアルのタイトル、バージョン、章タイトル、セクションタイトルも合わせてご報告ください。ご意見は次のアドレスに送信してください。

[NB.docs@veritas.com](mailto:NB.docs@veritas.com)

次のベリタスコミュニティサイトでマニュアルの情報を参照したり、質問することもできます。

<http://www.veritas.com/community/ja>

## ベリタスの Service and Operations Readiness Tools (SORT) の表示

ベリタスの Service and Operations Readiness Tools (SORT) は、時間がかかる管理タスクを自動化および簡素化するための情報とツールを提供する **Web** サイトです。製品によって異なりますが、SORT はインストールとアップグレードの準備、データセンターにおけるリスクの識別、および運用効率の向上を支援します。SORT がお客様の製品に提供できるサービスとツールについては、次のデータシートを参照してください。

[https://sort.veritas.com/data/support/SORT\\_Data\\_Sheet.pdf](https://sort.veritas.com/data/support/SORT_Data_Sheet.pdf)

# 目次

|       |                                                                 |    |
|-------|-----------------------------------------------------------------|----|
| 第 1 章 | NetBackup Parallel Streaming Framework を使<br>用したワークロードの保護 ..... | 7  |
|       | NetBackup Parallel Streaming Framework について .....               | 7  |
|       | Parallel Streaming Framework SDK について .....                     | 10 |
|       | Parallel Streaming Framework を使用したバックアップ処理 .....                | 12 |
|       | Parallel Streaming Framework を使用したリストア処理 .....                  | 13 |
|       | NetBackup Parallel Streaming Framework の用語 .....                | 14 |
| 第 2 章 | NetBackup Parallel Streaming Framework プラグ<br>インアーキテクチャ .....   | 16 |
|       | プラグインの操作 .....                                                  | 16 |
|       | NetBackup Parallel Streaming Framework API アーキテクチャ .....        | 17 |
|       | サポート対象の NetBackup 機能 .....                                      | 20 |
| 第 3 章 | NetBackup Parallel Streaming Framework SDK<br>の配備 .....         | 22 |
|       | オペレーティングシステムとプラットフォームの互換性 .....                                 | 22 |
|       | NetBackup Parallel Streaming Framework SDK のライセンス .....         | 23 |
|       | NetBackup Parallel Streaming Framework SDK のダウンロード .....        | 23 |
|       | NetBackup Parallel Streaming Framework SDK のインストール .....        | 24 |
| 第 4 章 | プラグイン開発の準備 .....                                                | 26 |
|       | NetBackup 環境の準備 .....                                           | 26 |
|       | バックアップホストの管理 .....                                              | 27 |
|       | NetBackup マスターサーバー上の NetBackup クライアントのホワイト<br>リスト .....         | 29 |
|       | BigData バックアップポリシーの作成 .....                                     | 29 |
|       | 認証機構の処理 .....                                                   | 33 |
|       | NetBackup での作業負荷クレデンシャルの追加 .....                                | 33 |
|       | セキュリティの考慮事項 .....                                               | 34 |
|       | プラグインを利用できるようにするために必要な認証 .....                                  | 35 |

|       |                                                            |    |
|-------|------------------------------------------------------------|----|
| 第 5 章 | サンプルプラグインの使用 .....                                         | 36 |
|       | サンプルプラグインについて .....                                        | 36 |
|       | サンプルプラグインの開発 .....                                         | 37 |
| 第 6 章 | 作業負荷プラグインの開発 .....                                         | 39 |
|       | プラグインの開発フェーズについて .....                                     | 39 |
|       | プラグインの初期化フェーズについて .....                                    | 41 |
|       | プラグインの要求フェーズについて .....                                     | 42 |
|       | プラグインのアンロードフェーズについて .....                                  | 42 |
|       | 作業負荷検出フェーズについて .....                                       | 43 |
|       | 作業負荷ファイルについて .....                                         | 47 |
|       | データバックアップフェーズについて .....                                    | 49 |
|       | リストアフェーズについて .....                                         | 52 |
|       | リカバリフェーズについて .....                                         | 56 |
|       | API の実行時間の管理 .....                                         | 58 |
|       | 開発中のプラグインの検証およびテスト .....                                   | 59 |
|       | 作業負荷プラグインのパッケージ化 .....                                     | 59 |
| 第 7 章 | ワークロードプラグインの配備 .....                                       | 61 |
|       | NetBackup File Checksum Validator ツールを使用したプラグインの検証 .....   | 61 |
|       | バックアップホストへの作業負荷プラグインのインストール .....                          | 63 |
|       | 作業負荷向けの NetBackup の構成 .....                                | 64 |
|       | 作業負荷のバックアップとリストアの実行 .....                                  | 64 |
| 第 8 章 | NetBackup Parallel Streaming Framework API<br>リファレンス ..... | 66 |
|       | aapi_close_object .....                                    | 67 |
|       | aapi_close_server .....                                    | 68 |
|       | aapi_create_object .....                                   | 69 |
|       | aapi_delete_object .....                                   | 70 |
|       | aapi_discovery .....                                       | 71 |
|       | aapi_flush .....                                           | 73 |
|       | aapi_get_object_prop .....                                 | 74 |
|       | aapi_get_object_prop_byname .....                          | 75 |
|       | aapi_get_server_prop .....                                 | 76 |
|       | aapi_get_snappath_from_realpath .....                      | 77 |
|       | aapi_open_object .....                                     | 78 |
|       | aapi_open_server .....                                     | 79 |
|       | aapi_pgn_claim .....                                       | 81 |

|       |                                                       |    |
|-------|-------------------------------------------------------|----|
|       | aapi_pgn_init .....                                   | 82 |
|       | aapi_postprocess .....                                | 83 |
|       | aapi_read_object .....                                | 84 |
|       | aapi_set_object_utimes .....                          | 86 |
|       | aapi_terminate .....                                  | 87 |
|       | aapi_write_object .....                               | 87 |
|       | aapi_restore_preprocess .....                         | 89 |
|       | aapi_restore_postprocess .....                        | 89 |
|       | NetBackup Parallel Streaming Framework API の戻り値 ..... | 90 |
| 第 9 章 | ログとトラブルシューティング .....                                  | 92 |
|       | プラグインのログの有効化 .....                                    | 92 |
| 索引    | .....                                                 | 97 |

# NetBackup Parallel Streaming Framework を使用したワークロードの保護

この章では以下の項目について説明しています。

- [NetBackup Parallel Streaming Framework](#) について
- [Parallel Streaming Framework SDK](#) について
- [Parallel Streaming Framework](#) を使用したバックアップ処理
- [Parallel Streaming Framework](#) を使用したリストア処理
- [NetBackup Parallel Streaming Framework](#) の用語

## NetBackup Parallel Streaming Framework について

Parallel Streaming Framework (PSF) を使用すると、NetBackup を使用して BigData 作業負荷を保護できるようになります。

- バックアップのために、Parallel Streaming Framework では完全バックアップと増分バックアップを実行できるほか、スナップショットもサポートされます。バックアップ作成はより効率的ですが、Parallel Streaming Framework はデータブロックを並行してストリームして、作業負荷を分散します。

- Parallel Streaming Framework では、NetBackup のカタログ、ポリシー、アクティビティモニター、およびログフレームワークを使用して、NetBackup フレームワークの利点を活用できます。

---

**メモ:** リカバリの側面は、プラグイン開発者が処理する必要があります。リカバリは、SDK の現在のスコープには含まれません。

---

- Parallel Streaming Framework は、特定の作業負荷プラグインを使用すると利用できます。プラグインはエージェントレスで、クラスタノード上に占有域はありません。
- ベリタスでは、作業負荷のプラグインを定期的にリリースしています。ただし、Parallel Streaming Framework SDK を使用すると、プラグインが利用可能でない場合に、特定の BigData 作業負荷向けのプラグインを作成できます。
  - p.16 の「[プラグインの操作](#)」を参照してください。
  - p.14 の「[NetBackup Parallel Streaming Framework の用語](#)」を参照してください。

---

**メモ:** 現在、SDK を使用して作成されたプラグインは、NetBackup アプライアンスではサポートされません。

---

Parallel Streaming Framework の使用を開始する前に、関連する用語の定義を確認します。p.14 の「[NetBackup Parallel Streaming Framework の用語](#)」を参照してください。

次の図は、BigData 作業負荷がどのように保護されるかの概要を示します。

---

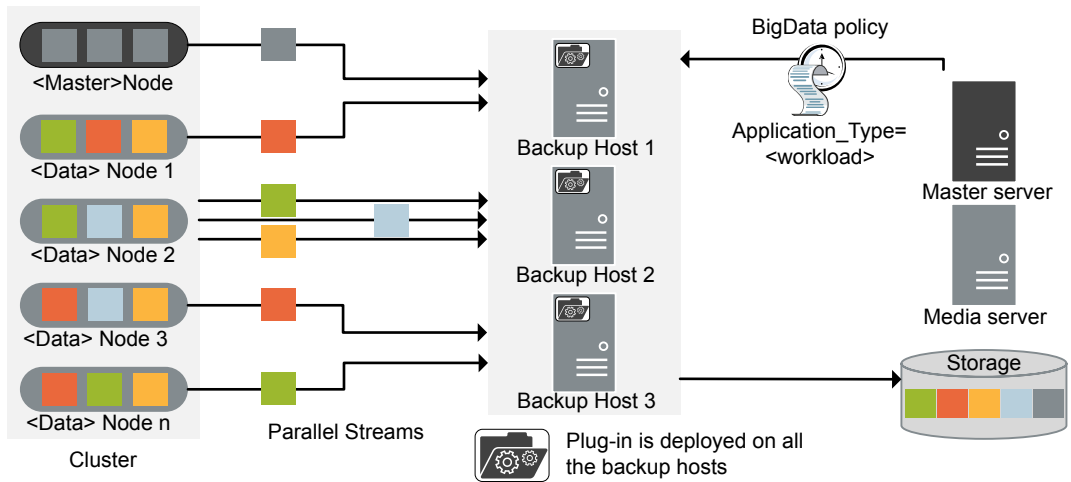
**メモ:** <Master> ノードは、アプリケーションソフトウェアとアプリケーションサーバーを実行するエンティティです。たとえば、Nutanix の場合はコントローラ VM で、Hadoop の場合はマスターノードで、HBase の場合は HMaster というようになります。

<Data> ノードは、バックアップが必要なエンティティです。たとえば、Nutanix の場合はゲスト VM で、Hadoop の場合はデータノードで、HBase の場合は地域サーバーというようになります。

---



図 1-1 アーキテクチャの概要



図では次の内容を説明しています。

- データは並列ストリームでバックアップされ、バックアップ時に **<Master>** ノードはデータブロックを同時に複数のバックアップホストに対してストリームします。ジョブの処理速度が、複数のバックアップホストと並列ストリームによって向上します。
- 作業負荷クラスと **NetBackup** 間の通信は、その作業負荷の **NetBackup** プラグインを使用して有効になります。  
プラグインは、すべてのバックアップホストにインストールする必要があります。
- **NetBackup** 通信の場合、**BigData** ポリシーを構成し、関連するバックアップホストを追加する必要があります。
- **NetBackup** のメディアサーバー、クライアント、またはマスターサーバーをバックアップホストとして構成することができます。また、**<Data>** ノードサーバーの数によっては、バックアップホストを追加または削除できます。バックアップホストをさらに追加することによって使用環境の規模を簡単に拡大することができます。
- **NetBackup** 並列ストリームフレームワークにより、エージェントレスのバックアップが可能で、バックアップとリストア操作はバックアップホストで実行します。クラスタノードには、エージェントの占有域がありません。また、**NetBackup** は & クラスタのアップグレードやメンテナンスの影響を受けません。

詳細情報:

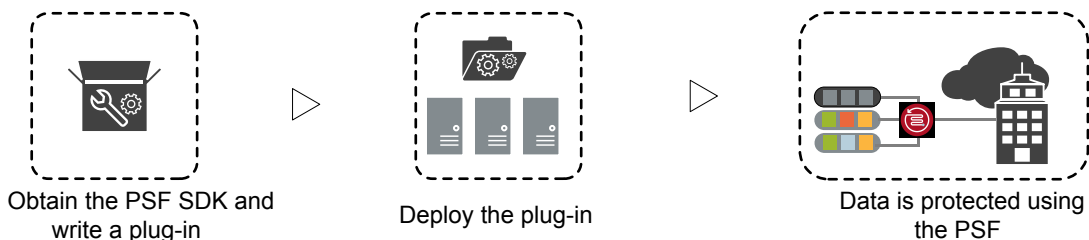
## Parallel Streaming Framework SDK について

Parallel Streaming Framework SDK は C ヘッダーファイルで構成されます。これらを使用すると、作業負荷プラグインを記述できます。プラグインを記述するときに、SDK で利用できるサンプルのプラグインを参照できます。

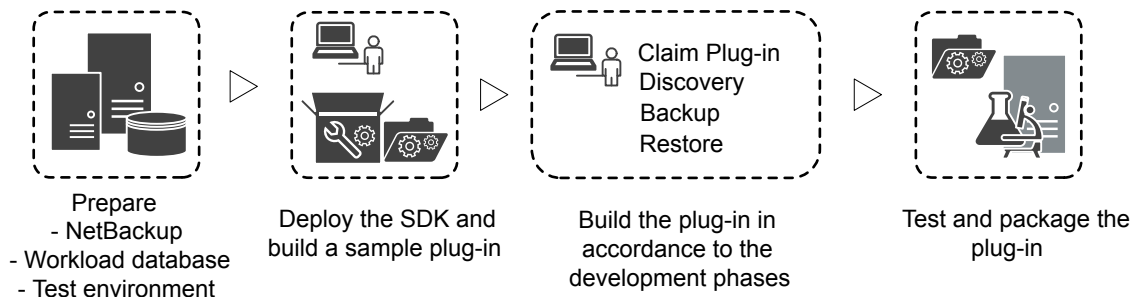
次の図は、Parallel Streaming Framework SDK の簡単な概要を示します。

図 1-2 Parallel Streaming Framework SDK の概要

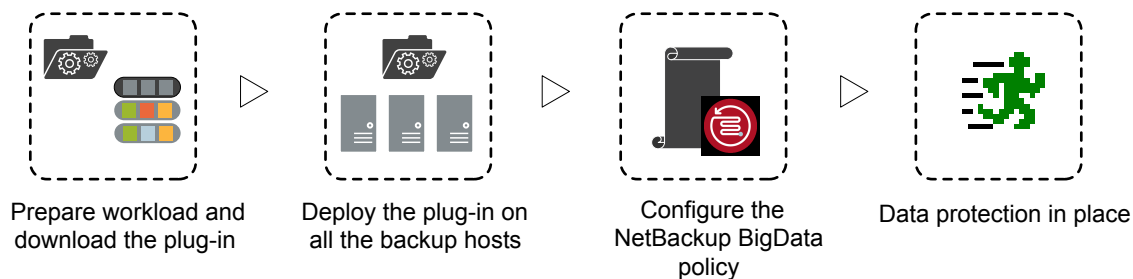
### Overview



### Plug-in development



### Plug-in deployment

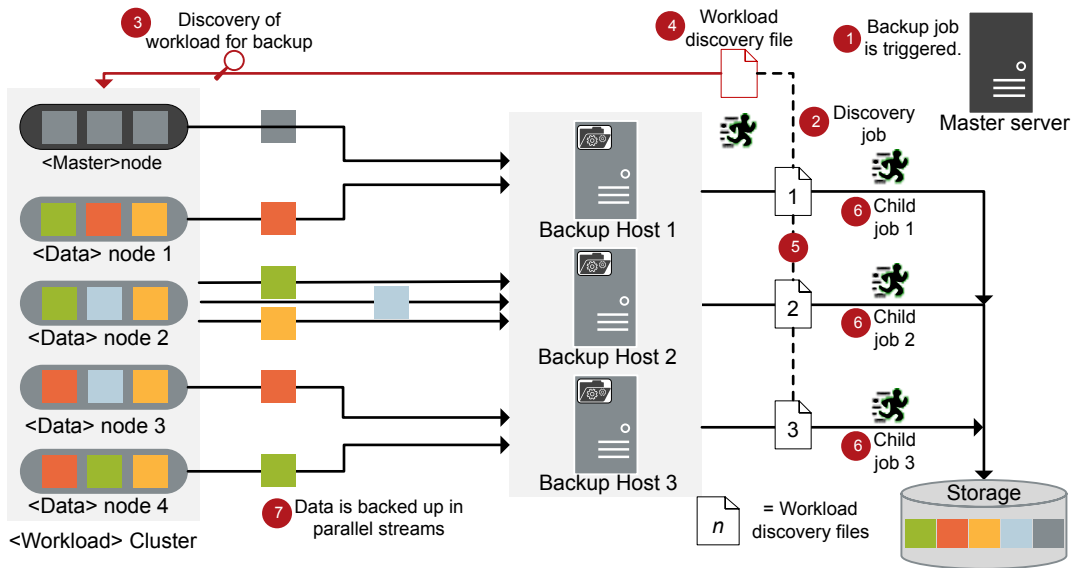


# Parallel Streaming Framework を使用したバックアップ処理

データは並列ストリームでバックアップされ、バックアップ時に <Data> ノードサーバーはデータブロックを同時に複数のバックアップホストに対してストリームします。

次の図は、バックアップ処理の概要を示しています。

図 1-3 バックアップ処理



次の図に示されているようになります。

1. スケジュールされたバックアップジョブはマスターサーバーからトリガされます。
2. バックアップジョブは、複合ジョブです。バックアップジョブがトリガされると、最初に検出ジョブが実行されます。
3. 検出中に、最初のバックアップホストは <Master> ノードと接続し、検出を実行して、バックアップする必要があるデータの詳細を取得します。
4. 作業負荷検出ファイルは、バックアップホストに作成されます。作業負荷検出ファイルには、さまざまな <Data> ノードサーバーからバックアップする必要があるデータの詳細が含まれています。
5. プラグインは作業負荷検出ファイルを使用し、作業負荷が複数のバックアップホスト間でどのように分散されるかを決定します。作業負荷検出ファイルは、バックアップホストごとに作成されます。

6. バックアップホストごとに個別の子ジョブが実行されます。作業負荷検出ファイルで指定されたデータがバックアップされます。
7. データブロックは、異なる <Data> ノードサーバーから複数のバックアップホストに同時にストリームします。

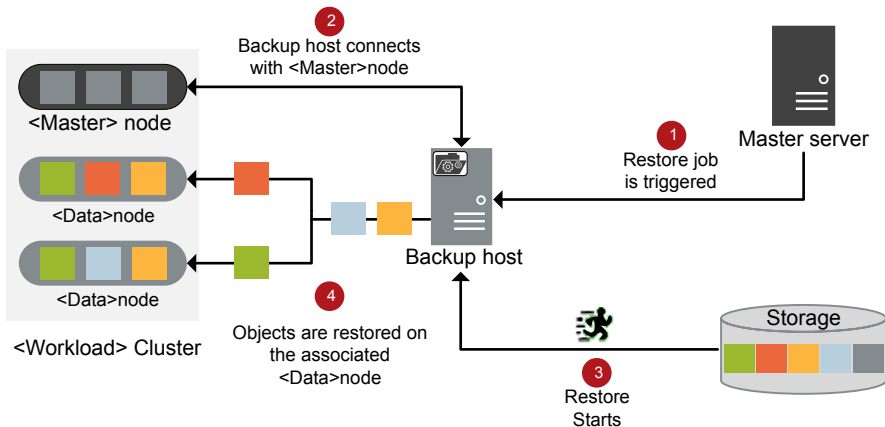
すべての子ジョブが完了するまで、複合バックアップジョブは完了しません。子ジョブが完了すると、NetBackup は <Master> ノードからすべてのスナップショットをクリーンアップします。クリーンアップ活動が完了した後にのみ、複合バックアップジョブは完了します。

## Parallel Streaming Framework を使用したリストア処理

リストアに使用されるのは、1 つのバックアップホストのみです。

次の図は、リストア処理の概要を示しています。

図 1-4 リストアフロー



図では次の内容を説明しています。

1. マスターサーバーからのリストアジョブがトリガされます。
2. バックアップホストが <Master> ノードと接続します。バックアップホストは宛先クライアントでもあります。
3. ストレージメディアからの実際のデータリストアが開始されます。

- データブロックは <Data> ノードサーバーにリストアされます。

# NetBackup Parallel Streaming Framework の用語

次の表では、BigData 作業負荷の保護に NetBackup を使用するときに使われる用語を定義しています。

表 1-1 Parallel Streaming Framework の用語

| 用語           | 定義                                                                                                                                                                                                                                                                                                   |
|--------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <Master> ノード | <Master> ノードは、アプリケーションソフトウェアとアプリケーションサーバーを実行するエンティティです。たとえば、Nutanix の場合はコントローラ VM で、Hadoop の場合はマスターノードで、HBase の場合は HMaster というようになります。                                                                                                                                                               |
| <Data> ノード   | <Data> ノードは、バックアップが必要なエンティティです。たとえば、Nutanix の場合はゲスト VM で、Hadoop の場合はデータノードで、HBase の場合は地域サーバーというようになります。                                                                                                                                                                                              |
| 複合ジョブ        | Parallel Streaming Framework を使用したバックアップジョブは、複合ジョブです。 <ul style="list-style-type: none"> <li>バックアップジョブは、バックアップするデータの情報を取得するための検出ジョブを実行します。</li> <li>子ジョブは、実際のデータ転送を実行する各バックアップホストに対して作成されます。</li> <li>バックアップが完了すると、ジョブは &lt;Master&gt; ノード上のスナップショットをクリーンアップし、その後ジョブ自体に完了したというマークが付けられます。</li> </ul> |
| 検出ジョブ        | バックアップジョブを実行すると、最初に検出ジョブが作成されます。検出ジョブは <Master> ノードと通信し、バックアップする必要があるブロックの情報と、関連する <Data> ノードサーバーの情報を収集します。検出の最後に、ジョブは作業負荷検出ファイルにデータを入力します。ファイルはその後 NetBackup によってバックアップホスト間で作業負荷を分散させるために使用されます。                                                                                                    |
| 子ジョブ         | バックアップの場合、ストレージメディアにデータを転送するバックアップホストごとに個別の子ジョブが作成されます。子ジョブは、複数の <Data> ノードサーバーからデータブロックを転送できます。                                                                                                                                                                                                     |
| 作業負荷検出ファイル   | 検出時のバックアップホストが <Master> ノードと通信するときに、作業負荷検出ファイルが作成されます。このファイルには、バックアップするデータブロックと、関連付けられている <Data> ノードサーバーについての情報が含まれています。                                                                                                                                                                             |

| 用語                    | 定義                                                                                                                                                                       |
|-----------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 並列ストリーム               | NetBackup 並列ストリームフレームワークにより、複数のバックアップホストを同時に使用して、複数の <Data> ノードサーバーからのデータブロックをバックアップできます。                                                                                |
| バックアップホスト             | すべてのバックアップとリストア操作は、バックアップホストで実行されます。<br>メディアサーバー、クライアント、またはマスターサーバーを、バックアップホストとして構成できます。<br>バックアップホストは、リストア中に宛先クライアントとしても使用されます。                                         |
| BigData ポリシー          | BigData ポリシーは以下を実行するために導入されました。 <ul style="list-style-type: none"> <li>■ アプリケーションの種類を指定します。</li> <li>■ 分散マルチノード環境のバックアップを可能にします。</li> <li>■ バックアップホストを関連付けます。</li> </ul> |
| アプリケーションサーバー          | <Master> ノードは、NetBackup ではアプリケーションサーバーと呼ばれます。                                                                                                                            |
| プライマリ <Master> ノード    | 高可用性シナリオでは、1 つの <Master> ノードを BigData ポリシーと tpconfig コマンドで指定する必要があります。この <Master> ノードは、プライマリ <Master> ノードと呼ばれます。                                                         |
| フェールオーバー <Master> ノード | 高可用性シナリオでは、プライマリ <Master> ノード以外の <Master> ノードです。                                                                                                                         |

# NetBackup Parallel Streaming Framework プラグインアーキテクチャ

この章では以下の項目について説明しています。

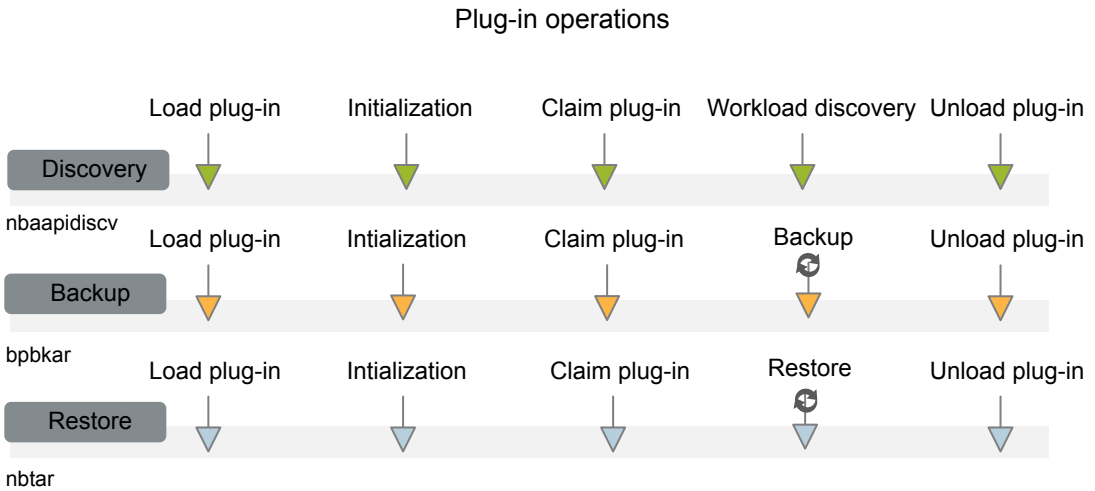
- [プラグインの操作](#)
- [NetBackup Parallel Streaming Framework API アーキテクチャ](#)
- [サポート対象の NetBackup 機能](#)

## プラグインの操作

次の図は、プラグイン操作の各フェーズのコンテキストでのプラグインアーキテクチャを示しています。PSF SDK を使用してプラグインを開発するときは、フェーズを意識し、それに応じた開発戦略を定義します。



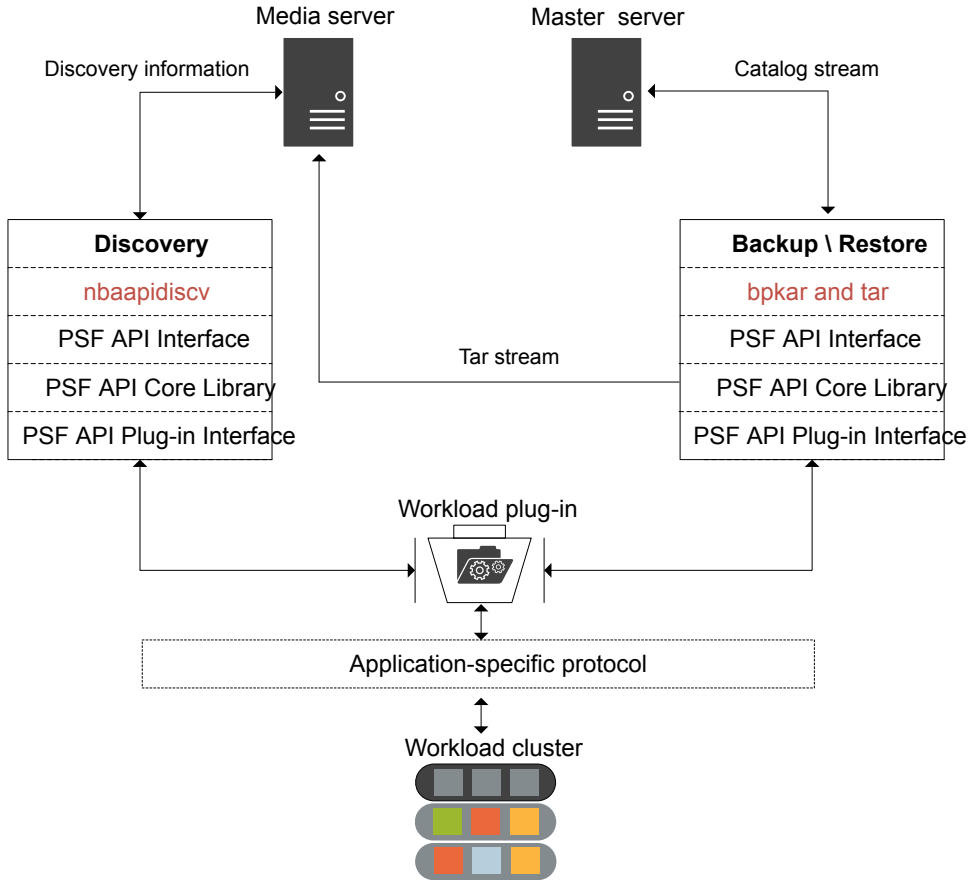
図 2-1 アーキテクチャの概要



## NetBackup Parallel Streaming Framework API アーキテクチャ

次の図に、NetBackup Parallel Streaming Framework API アーキテクチャを示します。

図 2-2 NetBackup Parallel Streaming Framework API アーキテクチャ



## NetBackup Parallel Streaming Framework API インターフェース

NetBackup は、Parallel Streaming Framework API を呼び出して Parallel Streaming Framework API コアライブラリにアクセスします。コアライブラリは、アプリケーションの種類または名前に基づいてプラグインを選択し、プラグインの呼び出しを発行します。NetBackup がアプリケーションと直接連携することはありません。

## NetBackup Parallel Streaming Framework API プラグインのインターフェース

プラグイン開発者は、プラグイン API が提供するインターフェースの必要なサブセットを実装する必要があります。プラグインは、オプションのインターフェースを実装する場合もしない場合もあります。オプションのインターフェースの実装は、アプリケーションの機能に依存します。

### NetBackup Parallel Streaming Framework API プラグイン

プラグインは、Parallel Streaming Framework API プラグインインターフェースに従って開発されます。プラグインは、API に必要なインターフェースを実装し、アプリケーションの機能に従ってオプションのインターフェースのサブセットを実装します。たとえば、スナップショットをサポートしないアプリケーションの場合、プラグインはスナップショットのインターフェースを実装しない可能性があります。プラグインの特徴を次に示します。

- プラグインは、アプリケーションのインターフェースとして機能します。
- プラグインは、アプリケーションの種類を要求します。
- プラグインは、アプリケーションのインターフェースとして機能します。アプリケーションは、プラグインと同じホスト上で実行したり、リモートホストで実行されているアプリケーションのインターフェースとして機能したりできます。

Parallel Streaming Framework API のコアライブラリとプラグインの間のすべてのやり取りは、Parallel Streaming Framework API インターフェースを通じて行われます。プラグインは、アプリケーション固有のプロトコルを使用してアプリケーションと通信します。たとえば、MongoDB データベースオブジェクトは、ネイティブな MongoDB インターフェースを使用して検出します。MongoDB プラグインは、その実装内で MongoDB に固有のすべてのロジックを非表示にし、統一された Parallel Streaming Framework API インターフェースを通じて機能を開示できます。

### NetBackup Parallel Streaming Framework API コアライブラリ

NetBackup は、Parallel Streaming Framework API コアライブラリを使用して、プラグインと連携します。Parallel Streaming Framework API コアライブラリは、複数のサービスを NetBackup に提供します。次に、主なサービスの一部を示します。

- アプリケーションの種類に基づいてプラグインを選択します。
- バックアップエンジンは、すべてのアプリケーションオブジェクトの読み取りを Parallel Streaming Framework API コアライブラリにリダイレクトします。  
Parallel Streaming Framework API コアライブラリは、アプリケーションの種類に基づいて特定のエージェントプラグインに読み取りをリダイレクトします。
- リストアエンジンは、すべてのアプリケーションオブジェクトの書き込みを Parallel Streaming Framework API コアライブラリにリダイレクトします。

Parallel Streaming Framework API コアライブラリは、アプリケーションの種類に基づいて特定のプラグインに読み取りをリダイレクトします。

- アプリケーションの静止サービスと静止解除サービス
- アプリケーションのオンラインサービスとオフラインサービス。このインターフェースが提供する機能を使用すると、エンジンがアプリケーションを停止したり開始したりできます。この機能は、アプリケーション回復中に必要です。
- スナップショットエンジンは、Parallel Streaming Framework API サービスを使用して、アプリケーションの論理オブジェクトを基になるストレージにマップできます。
- Oracle などの新しいアプリケーションの多くに、レプリケーション機能が組み込まれています。NetBackup プラットフォームは、Parallel Streaming Framework API を使用して、アプリケーション対応レプリケーションを実行できます。このために、アプリケーションのバックアップ、リストア、およびディザスタリカバリ用の単一の管理プラットフォームを提供します。

## ポリシーとアプリケーションの種類

Parallel Streaming Framework API アーキテクチャは、BigData という一般的なポリシーの種類を定義します。アプリケーションの種類 (たとえば、Hadoop または Nutanix) は、ポリシーの作成時に指定します。Parallel Streaming Framework API フレームワークを使用すると、NetBackup は、Parallel Streaming Framework API コアライブラリを通じて新しいアプリケーションプラグインを動的に検出します。このオプションにより、NetBackup は、リリースに関係なくアプリケーションをサポートできます。

# サポート対象の NetBackup 機能

作業負荷プラグインがサポートできるアプリケーションサーバーの機能は、`aapi_common.h` ヘッダーファイルで事前に定義されます。`aapi_common.h` ヘッダーファイルは、PSF SDK の一部です。

表 2-1 サポート対象の NetBackup 機能

| 機能                    | 説明                                                                       |
|-----------------------|--------------------------------------------------------------------------|
| AAPI_SRVC_CLAIM       | プラグインは <code>aapi_pgn_claim</code> API をサポートして、アプリケーションの種類をサポートします。      |
| AAPI_SRVC_TERMINATE   | プラグインは <code>aapi_terminate</code> API をサポートします。                         |
| AAPI_SRVC_OPEN_SERVER | プラグインは <code>aapi_open_server</code> API をサポートして、リモートアプリケーションサーバーに接続します。 |

| 機能                               | 説明                                                                         |
|----------------------------------|----------------------------------------------------------------------------|
| AAPI_SRVC_CLOSE_SERVER           | プラグインは <code>aapi_close_server</code> API をサポートして、アプリケーションサーバー接続を切断します。    |
| AAPI_SRVC_GET_SERVER_PROP        | プラグインは <code>aapi_get_server_prop</code> API をサポートして、プラグイン機能を公開します。        |
| AAPI_SRVC_GET_SERVER_PROP_BYNAME | プラグインは <code>aapi_get_server_prop_byname</code> API をサポートして、プラグイン機能を公開します。 |
| AAPI_SRVC_OPEN_OBJECT            | プラグインは <code>aapi_open_object</code> API をサポートして、バックアップするオブジェクトを開きます。      |
| AAPI_SRVC_CLOSE_OBJECT           | プラグインは <code>aapi_close_object</code> API をサポートして、開かれたオブジェクトを閉じます。         |
| AAPI_SRVC_CREATE_OBJECT          | プラグインは <code>aapi_create_object</code> API をサポートして、指定された場所にオブジェクトを作成します。   |
| AAPI_SRVC_DELETE_OBJECT          | プラグインは <code>aapi_delete_object</code> API をサポートして、オブジェクトを削除します。           |
| AAPI_SRVC_READ_OBJECT            | プラグインは <code>aapi_read_object</code> API をサポートして、開いているオブジェクトを読み込みます。       |
| AAPI_SRVC_WRITE_OBJECT           | プラグインは <code>aapi_write_object</code> API をサポートして、オブジェクトに書き込みます。           |
| AAPI_SRVC_FLUSH                  | プラグインは <code>aapi_flush</code> API をサポートして、オブジェクトデータをフラッシュします。             |
| AAPI_SRVC_GET_OBJECT_PROP        | プラグインは <code>aapi_get_object_prop</code> API をサポートして、オブジェクトプロパティを提供します。    |
| AAPI_SRVC_GET_OBJECT_PROP_BYNAME | プラグインは <code>aapi_get_object_prop</code> API をサポートして、オブジェクトプロパティを提供します。    |
| AAPI_SRVC_FULL_BACKUP_SUPPORT    | プラグインは、アプリケーションサーバーの完全バックアップをサポートします。                                      |
| AAPI_SRVC_INCR_BACKUP_SUPPORT    | プラグインは、アプリケーションサーバーの差分増分バックアップをサポートします。                                    |
| AAPI_SRVC_CINC_BACKUP_SUPPORT    | プラグインは、アプリケーションサーバーの累積増分バックアップをサポートします。                                    |

# NetBackup Parallel Streaming Framework SDK の配備

この章では以下の項目について説明しています。

- [オペレーティングシステムとプラットフォームの互換性](#)
- [NetBackup Parallel Streaming Framework SDK のライセンス](#)
- [NetBackup Parallel Streaming Framework SDK のダウンロード](#)
- [NetBackup Parallel Streaming Framework SDK のインストール](#)

## オペレーティングシステムとプラットフォームの互換性

NetBackup Parallel Streaming Framework SDK には、次のプラットフォームが必要です。

サポート対象のプラットフォーム:

- RHEL Server リリース 6.8 (Santiago) 以降
- SUSE Linux Enterprise Server 11 (x86\_64) (バージョン 11、パッチレベル 3) 以降

サポート対象のコンパイラ:

- RHEL: GCC バージョン 4.4.7 以降
- SUSE: GCC バージョン 4.3.4 以降

プラグインは 64 ビットである必要があります。

ここに示す要件は、サポートされる最小のバージョンです。プラグインを配備する環境の RHEL または SUSE のバージョンによっては、後のバージョンを使用できます。プラグインを開発する際は、glibc などの標準ライブラリの互換性を考慮します。

また、プラグインを開発するときに、リンクされているすべてのライブラリのバージョンが NetBackup のサポート対象であることを確認します。これは、プラグインでは後で NetBackup との連携が必要になるためです。

## NetBackup Parallel Streaming Framework SDK のライセンス

プラグインを開発するには、[VTE \(Veritas Technology Ecosystem\)](#) プログラムに登録する必要があります。

次のメンバーシップが必要です。

- VTE (Veritas Technology Ecosystem) プログラムのアクティブな基本メンバーシップ
- 米国の Technology Track のメンバーシップ

NetBackup でプラグインを使用するには、Enterprise Client ライセンスが必要です。

## NetBackup Parallel Streaming Framework SDK のダウンロード

ベリタスのサポートサイトから、次の NetBackup 用ファイルをダウンロードできます。

- `psf_sdk_UNIX.tar.gz`
- `README.checksum`: MD5、SHA-1、SHA-256 チェックサムが含まれています。

NetBackup Parallel Streaming Framework SDK プラグインをダウンロードするには

- 1 <https://partnernet.veritas.com/> ポータルに移動します。
- 2 サインインするには、ベリタスパートナープログラムのユーザー ID とパスワードを使用します。
- 3 [ライセンス]メニューから、NetBackup Parallel Streaming Framework SDK を特定してダウンロードします。
- 4 計画したバックアップホストのローカルディレクトリに、ダウンロードしたファイルを保存します。

# NetBackup Parallel Streaming Framework SDK のインストール

SDK をインストールして、gcc-c++ パッケージがインストールされた RHEL または SUSE 開発サーバー上にファイルを抽出する必要があります。開発サーバーへの NetBackup のインストールはオプションです。プラグインの開発に NetBackup は必要ありませんが、プラグインのテストには NetBackup が必要です。

NetBackup Parallel Streaming Framework SDK は、psf\_sdk/include ディレクトリに次のファイルを配置します。

表 3-1 SDK ファイル

| ファイル            | 目的                                                       |
|-----------------|----------------------------------------------------------|
| aapi_common.h   | プラグインの API が参照する共通ヘッダーファイルで、データ形式、データ構造、列挙、および型定義が含まれます。 |
| aapi_error.h    | プラグインが使用できる PSF エラーコード定義が含まれます。                          |
| aapi_platform.h | プラットフォーム固有のデータと型定義が含まれます。                                |
| aapi_plugin.h   | プラグインで使用される API のすべてのプロトタイプが含まれます。                       |

NetBackup Parallel Streaming Framework SDK は、psf\_sdk/plugins/sample ディレクトリに次のサンプルファイルを配置します。

表 3-2 サンプルプラグインのファイル

| ファイル                    | 目的                           |
|-------------------------|------------------------------|
| makefile                | サンプルプラグインをコンパイルしてビルドします。     |
| README                  | サンプルプラグインの構造について説明します。       |
| samplepgn_api.cpp       | すべてのフェーズのすべての主要機能の実装が含まれます。  |
| samplepgn_api.h         | すべての主要な関数の宣言です。              |
| samplepgn_discovery.cpp | 作業負荷ファイルを生成するための関数の実装が含まれます。 |



| ファイル                  | 目的                             |
|-----------------------|--------------------------------|
| samplepgn_discovery.h | 検出に使用する内部メソッドの宣言です。            |
| samplepgn_log.cpp     | NetBackup へのログ記録用の関数の実装が含まれます。 |
| samplepgn_log.h       | NetBackup へのログ記録用のすべての関数の宣言です。 |
| samplepgn_object.h    | 内部データ構造です。                     |
| samplepgn_types.h     |                                |

## 配置の検証

SDK の配置を検証するには、次の手順を完了します。

- plugins/sample フォルダに makefile が含まれることを確認します。
- ビルドホストが正しく構成されており、関連するすべてのパッケージがインストールおよび構成されていることを確認します。簡単なプログラムのビルドと実行を試みます。makefile のインクルードパスを確認します。  
makefile では、相対パスを使用して SDK のヘッダーファイルを参照します。インクルードフォルダが別の場所に移動されている場合は、makefile でも相応の変更が必要です。
- サンプルプラグインをビルドしてテストします。
  - 次のコマンドのいずれかを使用して、libaapipgnsample.so サンプルプラグインをビルドします。
    - make libaapipgnsample.so
    - make all
  - サンプルプラグインに、すべて (ユーザー、グループ、およびその他) に対する読み取りと実行などの適切な権限があることを確認します。
  - このサンプルプラグインを使用して、バックアップとリストアをテストします。
- サンプルプラグインに、すべて (ユーザー、グループ、およびその他) に対する読み取りと実行などの適切な権限があることを確認します。

# プラグイン開発の準備

この章では以下の項目について説明しています。

- [NetBackup 環境の準備](#)
- [バックアップホストの管理](#)
- [BigData バックアップポリシーの作成](#)
- [認証機構の処理](#)
- [NetBackup での作業負荷クレデンシャルの追加](#)
- [セキュリティの考慮事項](#)
- [プラグインを利用できるようにするために必要な認証](#)

## NetBackup 環境の準備

プラグインと NetBackup が連携するためには、次のタスクを完了する必要があります。

表 4-1 NetBackup 環境の準備

| 作業           | 説明                                                                                                                                                                                                                                                                                          |
|--------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| バックアップホストの設定 | <ul style="list-style-type: none"><li>■ 開発前:<br/>バックアップホストとして使用するサーバーを識別します。</li><li>■ テストおよび配備中:<ul style="list-style-type: none"><li>■ 各バックアップホストにプラグインをインストールします。</li><li>■ NetBackup ポリシーを使ってバックアップホストを追加します。</li></ul></li></ul> <p>p.27 の「<a href="#">バックアップホストの管理</a>」を参照してください。</p> |

| 作業                  | 説明                                                                                                                                                                                                                                                                                                                                                                                           |
|---------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| BigData ポリシーを構成します。 | <ul style="list-style-type: none"><li>■ 開発前:<ul style="list-style-type: none"><li>■ apptype 値を定義します。</li><li>■ バックアップへのパスを定義します。</li><li>■ バックアップホストを識別します。</li></ul></li><li>■ テストおよび配備中:<ul style="list-style-type: none"><li>■ apptype 値を構成します。</li><li>■ バックアップへのパスを指定します。</li><li>■ バックアップホストを追加します。</li></ul></li></ul> <p>p.29 の「<a href="#">BigData バックアップポリシーの作成</a>」を参照してください。</p> |
| 作業負荷クレデンシャルを更新します。  | <ul style="list-style-type: none"><li>■ 開発前:<br/>作業負荷のシンプルテキストのクレデンシャル (ユーザー名とパスワード) を取得します。</li><li>■ テストおよび配備中:<br/>tpconfig コマンドを使用してクレデンシャルを更新します。</li></ul> <p>p.33 の「<a href="#">NetBackup での作業負荷クレデンシャルの追加</a>」を参照してください。</p>                                                                                                                                                         |

## バックアップホストの管理

バックアップホストは、作業負荷クラスタのすべてのバックアップとリストア操作をホストするプロキシクライアントとして機能します。バックアップホストは、作業負荷クラスタにインストールされている独立したエージェントなしですべてのバックアップとリストア操作を実行します。

バックアップホストは、Linux コンピュータである必要があります。NetBackup は、バックアップホストとして RHEL および SUSE プラットフォームのみをサポートします。

バックアップホストとして、NetBackup クライアント、メディアサーバー、またはマスターサーバーを使用することができます。メディアサーバーをバックアップホストとして設定することをお勧めします。

バックアップホストを追加する前に、次の点を考慮します。

- バックアップ操作用に、1 つまたは複数のバックアップホストを追加できます。
- リストア操作用に、バックアップホストを 1 つだけ追加できます。
- プラグインは、すべてのバックアップホストにインストールされます。
- NetBackup クライアントをバックアップホストとして使用するには、それをホワイトリストに載せる必要があります。p.29 の「[NetBackup マスターサーバー上の NetBackup クライアントのホワイトリスト](#)」を参照してください。

**NetBackup 管理コンソールからバックアップホストを追加するには**

- 1 **BigData** ポリシーを作成します。p.29 の「**BigData** バックアップポリシーの作成」を参照してください。
- 2 **BigData** ポリシーを作成するときに、[バックアップ対象]タブをクリックします。
- 3 [バックアップ対象]タブでは、[新規]をクリックし、次の形式でバックアップホストを追加します。

**Backup\_Host=<IP アドレスまたはホスト名>**

また、次のコマンドを使用して、バックアップホストを追加することもできます。

**Windows** の場合:

```
bpplinclude PolicyName -add "Backup_Host=IP アドレスまたはホスト名"
```

**UNIX** の場合:

```
bpplinclude PolicyName -add 'Backup_Host=IP アドレスまたはホスト名'
```

- 4 ベストプラクティスとして、すべてのバックアップホスト上の /etc/hosts ファイルに、すべての **<Master>** ノードと **<Data>** ノードのエントリを追加します。FQDN 形式でホスト名を追加する必要があります。

または

/etc/resolve.conf ファイルに適切な DNS エントリを追加します。

**NetBackup コマンドラインインターフェースを使用してバックアップホストを追加するには**

- 1 **BigData** ポリシーを作成します。p.29 の「**BigData** バックアップポリシーの作成」を参照してください。
- 2 次のコマンドを使用して、バックアップホストを追加します。

**Windows** の場合:

```
bpplinclude PolicyName -add "Backup_Host=IP アドレスまたはホスト名"
```

**UNIX** の場合:

```
bpplinclude PolicyName -add 'Backup_Host=IP アドレスまたはホスト名'
```

- 3 ベストプラクティスとして、すべてのバックアップホスト上の /etc/hosts ファイルに、すべての **<Master>** ノードと **<Data>** ノードのエントリを追加します。FQDN 形式でホスト名を追加する必要があります。

または

/etc/resolve.conf ファイルに適切な DNS エントリを追加します。

バックアップホストを削除するには

- 1 [バックアップ対象]タブで、削除するバックアップホストを選択します。
- 2 選択したバックアップホストを右クリックし、[削除]をクリックします。  
また、次のコマンドを使用して、バックアップホストを削除することもできます。

Windows の場合:

```
bpplinclude PolicyName -delete "Backup_Host=IP アドレスまたはホスト名"
```

UNIX の場合:

```
bpplinclude PolicyName -delete 'Backup_Host=IP アドレスまたはホスト名'
```

## NetBackup マスターサーバー上の NetBackup クライアントのホワイトリスト

NetBackup クライアントをバックアップホストとして使用するには、それをホワイトリストに載せる必要があります。NetBackup マスターサーバー上でホワイトリストへの追加手順を実行します。

ホワイトリストは、ソフトウェアまたはアプリケーションが安全な実行を承認されていない限り、それらを実行しないようにシステムを制限するセキュリティ手法です。

NetBackup マスターサーバー上で NetBackup クライアントをホワイトリストに載せるには

- ◆ NetBackup マスターサーバー上で次のコマンドを実行します。

```
bpsetconfig -h masterserver
```

```
bpsetconfig> APP_PROXY_SERVER = clientname.domain.org
```

```
bpsetconfig>
```

UNIX システムの場合: <ctl-D>

Windows システムの場合: <Ctrl-Z>

このコマンドは `APP_PROXY_SERVER = clientname` エントリをバックアップ構成 (bp.conf) ファイルに設定します。

`APP_PROXY_SERVER = clientname` について詳しくは、『NetBackup 管理者ガイド Vol. 1』の NetBackup クライアントの構成オプションのセクションを参照してください。

## BigData バックアップポリシーの作成

ビッグデータアプリケーションをバックアップするには、BigData ポリシーを使用します。

BigData ポリシーは、他のポリシーとは次の点で異なります。

- ポリシー形式として **BigData** を指定する必要があります。
- [クライアント]タブと[バックアップ対象]で提供されるエントリは、バックアップを作成するアプリケーションに応じて異なります。
- [バックアップ対象]タブで、特定のパラメータとその適切な値を指定する必要があります。

NetBackup 管理コンソールまたはコマンドラインインターフェースのいずれかを使用して、BigData ポリシーを作成できます。

**NetBackup ポリシーユーティリティを使用して BigData ポリシーを作成するには**

- 1 NetBackup 管理コンソールの左ペインで、[NetBackup の管理 (NetBackup Management)]>[ポリシー (Policies)]を展開します。
- 2 [処理 (Actions)]メニューで[新規 (New)]>[ポリシー (Policy)]をクリックします。
- 3 新しいポリシー用の一意の名前を[新しいポリシーの追加 (Add a New Policy)]ダイアログボックスに入力します。  
[OK]をクリックします。

- 4 [属性 (Attributes)]タブで、ポリシー形式に[BigData]を選択します。
- 5 [属性 (Attributes)]タブには、BigData ポリシー形式のストレージユニットを選択します。
- 6 [スケジュール (Schedules)]タブで[新規 (New)]をクリックして、新しいスケジュールを作成します。

BigData ポリシーの完全バックアップ、差分増分バックアップ、または累積増分バックアップのスケジュールを作成できます。スケジュールを設定すると、<workload> データは、ユーザーがそれ以上介入しなくても、設定されたスケジュールに従って自動的にバックアップされます。

- 7 [クライアント (Clients)]タブには、NameNode の IP アドレスまたはホスト名を入力します。
- 8 [バックアップ対象 (Backup Selections)]タブで、次のようにパラメータとその値を入力します。

- **Application\_Type=<workload>**  
プラグインのサンプルに **aapisample** を指定します。  
これらのパラメータ値では、大文字と小文字が区別されます。
- **Backup\_Host=IP\_address or hostname**  
バックアップホストは、Linux コンピュータである必要があります。バックアップホストには、NetBackup クライアントまたはメディアサーバーを指定できます。  
複数のバックアップホストを指定できます。
- バックアップするエンティティ。

---

**メモ:** BigData ポリシーを `Application_Type=<workload>` で定義するときにバックアップ対象に対して指定されるエンティティには、名前にスペースまたはカンマを含めることはできません。

---

- 9 [OK]をクリックして、変更を保存します。

**NetBackup CLI 方式を使用して BigData ポリシーを作成するには**

- 1 管理者としてログオンします。
- 2 `/usr/openv/netbackup/bin/admincmd` に移動します。
- 3 デフォルト設定を使用して、新しい BigData ポリシーを作成します。

```
bpolicynew policyname
```

- 4 `-L` オプションを使用して、新しいポリシーの詳細を表示します。

```
bpplinfo policyname -L
```

- 5 ポリシー形式を BigData として変更および更新します。

```
bpplinfo PolicyName -modify -v -M MasterServerName -pt BigData
```

- 6 `Application_Type` を `<workload>` として指定します。

Windows の場合:

```
bpplinclude PolicyName -add "Application_Type=workload"
```

UNIX の場合:

```
bpplinclude PolicyName -add 'Application_Type=workload'
```

---

**メモ:** `Application_Type=workload` のパラメータ値では、大文字と小文字が区別されます。

---

- 7 作業負荷に対するバックアップ操作を実行するバックアップホストを指定します。

Windows の場合:

```
bpplinclude PolicyName -add "Backup_Host=IP アドレスまたはホスト名"
```

UNIX の場合:

```
bpplinclude PolicyName -add 'Backup_Host=IP アドレスまたはホスト名'
```

---

**メモ:** バックアップホストは、Linux コンピュータである必要があります。バックアップホストとして、NetBackup クライアント、メディアサーバー、またはマスターサーバーを使用することができます。

---

- 8 バックアップする作業負荷ディレクトリまたはフォルダ名を指定します。

Windows の場合:

```
bpplinclude PolicyName -add "/workloadfoldername"
```

UNIX の場合:

```
bpplinclude PolicyName -add '/workloadfoldername'
```

---

**メモ:** BigData ポリシーを `Application_Type=<workload>` で定義するときにバックアップ対象に使用されるディレクトリまたはフォルダには、名前に特殊文字を含めることはできません。

---

- 9 BigData ポリシーのポリシーストレージ形式を変更および更新します。

```
bpplinfo PolicyName -residence STUName -modify
```

- 10 クライアント詳細を追加するための、<Master> ノードの IP アドレスまたはホスト名を指定します。

Windows の場合:

```
bpplclients PolicyName -M "MasterServerName" -add  
"<workload>ServerNameNode" "Linux" "RedHat"
```

UNIX の場合:

```
bpplclients PolicyName -M 'MasterServerName' -add  
'<workload>ServerNameNode' 'Linux' 'RedHat'
```

- 11 要件に応じて作成された BigData ポリシーにスケジュールを割り当てます。

```
bpplsched PolicyName -add Schedule_Name -cal 0 -rl 0 -st  
sched_type -window 0 0
```

ここで、`sched_type` 値は次のように指定できます。

| スケジュール形式 | 説明         |
|----------|------------|
| FULL     | 完全バックアップ   |
| INCR     | 差分増分バックアップ |
| CINC     | 累積増分バックアップ |

`sched_type` のデフォルト値は **FULL** です。

スケジュールを設定すると、<workload> データは、ユーザーがそれ以上介入しなくても、設定されたスケジュールに従って自動的にバックアップされます。



**12** 別の方法として、<workload> データの手動バックアップを実行することもできます。  
手動バックアップ操作を実行するには、手順 1 から手順 11 のすべての手順を実行します。

**13** 手動バックアップ操作では、/usr/opensv/netbackup/bin に移動します。

次のコマンドを使用して、既存の **BigData** ポリシーの手動バックアップ操作を開始します。

```
bpbackup -i -p PolicyName -s Schedule_Name -S MasterServerName  
-t 44
```

ここで、-p はポリシー、-s はスケジュール、-S はマスターサーバー、および -t 44 は **BigData** ポリシー形式を表しています。

## 認証機構の処理

認証の実装では、次の点を考慮します。

- デフォルトでは、簡易認証のみが有効になっています (ユーザー ID とパスワード)。
- プラグインのプロバイダは、他の認証の種類 (**SSL**、**Kerberos** など) を処理する必要があります。たとえば、**Kerberos** の場合、バックアップを実行するために、バックアップホストに有効な **TGT** が必要です。
- 認証機構と作業負荷の種類によっては、tpconfig エントリを更新する必要があります。

## NetBackup での作業負荷クレデンシャルの追加

正常なバックアップとリストア操作のために作業負荷クラスタと **NetBackup** との間でシームレスな通信を確立するには、作業負荷クレデンシャルを **NetBackup** マスターサーバーに追加して更新します。

**NetBackup** マスターサーバーで クレデンシャルを追加するには、tpconfig コマンドを使用します。

tpconfig コマンドについて詳しくは、『[NetBackup コマンドリファレンスガイド](#)』を参照してください。

クレデンシャルを追加する場合は、次の点を考慮します。

- 可用性の高い作業負荷クラスタの場合、プライマリとフェールオーバーの **MasterNode** のユーザーが同じであることを確認します。
- **BigData** ポリシーを構成するときに、アプリケーションサーバーのクレデンシャルを使用します。

- **MasterNode** のホスト名とポートは、作業負荷クラスタの HTTP アドレスパラメータで指定したものと同じでなければなりません。

### NetBackup で作業負荷クレデンシャルを追加するには

- 1 次のディレクトリパスから `tpconfig` コマンドを実行します。

UNIX システムでは、`/usr/opensv/volmgr/bin/`

Windows システムでは、`install_path\Volmgr\bin\`

- 2 `tpconfig --help` コマンドを実行します。**<workload>** クレデンシャルを追加、更新、および削除するのに必要なオプションのリストが表示されます。

- 3 `tpconfig -add -application_server application_server_name -application_server_user_id user_ID -application_type application_type -requiredport IP_port_number [-password password [-key encryption_key]]` コマンドを各パラメータに適切な値を入力して実行し、作業負荷クレデンシャルを追加します。

たとえば、**application\_server\_name** が **workload1** である作業負荷サーバーのクレデンシャルを追加する場合、適切な **<user\_ID>** と **<password>** の詳細を使用して次のコマンドを実行します。

```
tpconfig -add -application_server <workload>1 -application_type  
1 -application_server_user_id workload -password workload
```

ここで、`-application_type` パラメータに指定された数値 1 は作業負荷に対応します。

- 4 `tpconfig -dappservers` コマンドを実行し、**NetBackup** マスターサーバーに追加された作業負荷クレデンシャルがあることを確認します。

## セキュリティの考慮事項

作業負荷プラグインを安全に配備するには、開発時に次の点を考慮します。

- プラグインのバイナリは、ルートのみアクセス権を使用して配備する必要があります。
- プラグインで、ベリタスのログフレームワークを使用する必要があります。プラグインの開発時に、ログについて十分な情報を追加する必要があります。
- 作業負荷ファイルの安全を確保するには、作業負荷ファイルを作成するときに、ルートのみ権限で **ACL** を設定する必要があります。
- **NetBackup** インストールパスにプラグインを配備するには、プラグインを事前定義済みの `/usr/opensv/lib/psf-plugins/third_party/plugin_name` パスに配備する必要があります。ここで、**plug-in name** をアプリケーションの種類と同じにすることを勧めます。

## プラグインを利用できるようにするために必要な認証

プラグインを開発したら、ベリタスと連携してプラグインを検証して承認を得る必要があります。プラグインの認証の一環として、ベリタスでは、推奨されるガイドラインに従ってプラグインが記述されているかどうかを確認します。検証プロセスの際に、プラグインの開発者は、ベリタスの仮想チームと基本的なテストを実行する必要があります。

プラグインの安全な配備に対する完全なチェックは、認証には含まれません。最適なセキュリティを使用してプラグインを記述することは、プラグイン開発者の責任になります。計画フェーズでは、ベリタスチームがセキュリティ SME と検証できるように、セキュリティガイドラインを策定します。

認証の詳細プロセスは、計画フェーズの際にベリタスチームと話し合うことができます。大まかに見ると、プロセスは次のタスクで構成されます。

- ベリタスチームが提供するテストケースを実行します。
- プラグインを公開する前に、検証とサインオフを受けるために、ベリタスチームにレポートを提出します。
- プラグインの認証中にベリタスチームが特に考慮する必要がある事項があるかどうかを通知します。
- チェックサムを検証するために、プラグインをビルドして配備したコンピュータにベリタスチームがアクセスする必要がある場合があります。
- プラグインの認定後、ベリタスチームは **NetBackup** ソフトウェア互換性リストにプラグインのサポートを追加し、ベリタスのセキュアサイトにチェックサムが発行されます。
- **NetBackup** またはプラグインに主要な変更が発生した場合に、再認証の開始が必要になることがあります。

# サンプルプラグインの使用

この章では以下の項目について説明しています。

- サンプルプラグインについて
- サンプルプラグインの開発

## サンプルプラグインについて

プラグイン開発を実践的に体験するために、サンプルプラグインを参考にしながら、基本的な機能を含めたサンプルプラグインを構築できます。

サンプルプラグインには、このバージョンの SDK でサポートされる機能のみが含まれます。

サンプルプラグインは、単純なファイルシステムをバックアップする目的で設計されています。単純なファイルシステムのデータは配布できないため、サンプルプラグインでは複数のバックアップホストはサポートされません。

バックアップホスト上に存在するデータのバックアップを作成する必要があります。

---

**メモ:** 開始する前に、開発フェーズを理解し、フェーズに従って続行します。

---

## サンプルプラグインの内容

サンプルプラグインには、次のファイルが含まれます。

**表 5-1**                      サンプルプラグインのファイル

| ファイル     | 目的                       |
|----------|--------------------------|
| makefile | サンプルプラグインをコンパイルしてビルドします。 |
| README   | サンプルプラグインの構造について説明します。   |

| ファイル                    | 目的                             |
|-------------------------|--------------------------------|
| samplepgn_api.cpp       | すべてのフェーズのすべての主要機能の実装が含まれます。    |
| samplepgn_api.h         | すべての主要な関数の宣言です。                |
| samplepgn_discovery.cpp | 作業負荷ファイルを生成するための関数の実装が含まれます。   |
| samplepgn_discovery.h   | 検出に使用する内部メソッドの宣言です。            |
| samplepgn_log.cpp       | NetBackup へのログ記録用の関数の実装が含まれます。 |
| samplepgn_log.h         | NetBackup へのログ記録用のすべての関数の宣言です。 |
| samplepgn_object.h      | 内部データ構造です。                     |
| samplepgn_types.h       |                                |

## 関連項目

p.37 の「[サンプルプラグインの開発](#)」を参照してください。

# サンプルプラグインの開発

このセクションでは、サンプルプラグインの使用方法を説明するための例として、検出フェーズを使用します。同様に、すべてのフェーズの開発に同じ手順を使用できます。

次の表は、検出の関数とサンプルの実装の対応を示します。

表 5-2 検出の関数とサンプルの実装

| SDK の関数               | サンプルの実装 (samplepgn_api.cpp) |
|-----------------------|-----------------------------|
| aapi_pgn_open_server  | samplepgn_open_server_v1    |
| aapi_pgn_init         | aapi_pgn_init               |
| aapi_pgn_claim        | samplepgn_claim_v1          |
| aapi_pgn_discovery    | samplepgn_discovery_v1      |
| aapi_pgn_close_server | samplepgn_close_server_v1   |
| aapi_pgn_terminate    | samplepgn_terminate_v1      |

### サンプルプラグインを開発するには

- 1 ヘッダー (.h) およびソース (.c または .cpp) ファイルを作成します。
- 2 ヘッダーファイル内で関数を宣言します。
- 3 宣言した関数をソースファイルに実装します。
- 4 ログ記録を実装していることを確認します。
- 5 作成したプラグイン向けに次の情報を使用して **make** ファイルを更新します。
  - ソースファイルの場所
  - SDK のインクルードパス
  - プラグインのヘッダーファイルのパス
  - 依存ライブラリを使用した **CXXFLAGS**
  - 出力ライブラリ (.so) ファイル
- 6 次のコマンドのいずれかを使用して、libaapipgnsample.so サンプルプラグインをビルドします。
  - `make libaapipgnsample.so`
  - `make all`
- 7 出力ライブラリ (.so ファイル) が出力ディレクトリに作成されるかどうかを確認します。
- 8 プラグインライブラリへの実行アクセス権を提供します。
- 9 **NetBackup** マスターサーバー上に **BigData** ポリシーを作成します。
- 10 `tpconfig` コマンドを実行して、アプリケーションサーバーとアプリケーションの種類を追加します。
- 11 バックアップジョブを実行して、検出ジョブが完了したかどうかを確認します。
- 12 `bpbrm` と `nbaapidiscv` のログファイルを確認して、ジョブの実行を検証します。

# 作業負荷プラグインの開発

この章では以下の項目について説明しています。

- プラグインの開発フェーズについて
- プラグインの初期化フェーズについて
- プラグインの要求フェーズについて
- プラグインのアンロードフェーズについて
- 作業負荷検出フェーズについて
- データバックアップフェーズについて
- リストアフェーズについて
- リカバリフェーズについて
- API の実行時間の管理
- 開発中のプラグインの検証およびテスト
- 作業負荷プラグインのパッケージ化

## プラグインの開発フェーズについて

次の表では、プラグイン開発のフェーズをおおまかに分類しています。各フェーズは、作業負荷を保護するためにプラグインで実行するさまざまな操作に合わせられています。

表 6-1 プラグインの要求フェーズと API 呼び出しの順番

| フェーズ       | 参照先                                                                                                                                                                                                                   |
|------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| プラグインの共通操作 | 検出、バックアップ、リストアのために次のフェーズが実行されます。<br><br><a href="#">p.41 の「プラグインの初期化フェーズについて」</a> を参照してください。<br><br><a href="#">p.42 の「プラグインの要求フェーズについて」</a> を参照してください。<br><br><a href="#">p.42 の「プラグインのアンロードフェーズについて」</a> を参照してください。 |
| 作業負荷の検出    | <a href="#">p.43 の「作業負荷検出フェーズについて」</a> を参照してください。                                                                                                                                                                     |
| データバックアップ  | <a href="#">p.49 の「データバックアップフェーズについて」</a> を参照してください。                                                                                                                                                                  |
| データリストア    | <a href="#">p.52 の「リストアフェーズについて」</a> を参照してください。                                                                                                                                                                       |

表 6-2 他の開発タスク

| 作業                  | 参照先                                                                                                        |
|---------------------|------------------------------------------------------------------------------------------------------------|
| NetBackup へのエラー報告   | プラグイン操作に関連するエラーは、プラグインを使用するすべてのログファイルに記録される必要があります。<br><br><a href="#">p.92 の「プラグインのログの有効化」</a> を参照してください。 |
| SDK の問題のトラブルシューティング |                                                                                                            |
| API の実行時間の管理        | <a href="#">p.58 の「API の実行時間の管理」</a> を参照してください。                                                            |
| 開発中のプラグインの検証およびテスト  | <a href="#">p.59 の「開発中のプラグインの検証およびテスト」</a> を参照してください。                                                      |
| プラグインのパッケージ化        | <a href="#">p.59 の「作業負荷プラグインのパッケージ化」</a> を参照してください。                                                        |



表 6-3 開始する前に

| 作業            | 参照先                                               |
|---------------|---------------------------------------------------|
| アーキテクチャの理解    | p.16 の「 <a href="#">プラグインの操作</a> 」を参照してください。      |
| プラグインの配備      | <a href="#">第3章</a>                               |
| 開発前タスクの完了     | <a href="#">第4章</a>                               |
| サンプルプラグインのビルド | p.36 の「 <a href="#">サンプルプラグインについて</a> 」を参照してください。 |

## プラグインの初期化フェーズについて

プラグインの初期化は、検出、バックアップ、およびリストアの 3 つの使用例すべてで最初に実行するフェーズです。

このフェーズでは、SDK バージョンが確認されて、対応するメソッドテーブルが設定されます。メソッドテーブルには、プラグインで使用するすべての API のリストがあります。

### 関連するプロセスと API

bpbrm プロセスは、メディアサーバー上で実行されます。

プラグインの初期化フェーズでは、次の API が順次呼び出されます。

表 6-4 プラグインの初期化フェーズ API 呼び出しの順番

| フェーズ                | API 呼び出し                         | サンプルのプラグインの参照               |
|---------------------|----------------------------------|-----------------------------|
| プラグインの初期化           | <code>aapi_pgn_init</code>       | aapi_pgn_init               |
| アプリケーションサーバーとの接続の設定 | <code>aapi_open_server</code>    | samplepgn_open_server_v1    |
| プラグインのアンロード         | 1 <code>aapi_close_server</code> | 1 samplepgn_close_server_v1 |
|                     | 2 <code>aapi_terminate</code>    | 2 samplepgn_terminate_v1    |

### NetBackup のプラグインの初期化フェーズでのログ情報の追加

既存のエラーとは別に、関連するデバッグログファイルにカスタムログを作成します。

p.92 の「[プラグインのログの有効化](#)」を参照してください。

### 関連項目

p.59 の「[開発中のプラグインの検証およびテスト](#)」を参照してください。

p.36 の「[サンプルプラグインについて](#)」を参照してください。

## プラグインの要求フェーズについて

プラグインの初期化後、検出、バックアップ、およびリストアの 3 つの使用例すべてで、プラグインの要求フェーズが実行されます。

このフェーズでは、アプリケーションの種類 (apptype) を使用して、NetBackup が特定の作業負荷に対するプラグインの可用性を確認します。

### 関連するプロセスと API

bpbrm プロセスは、メディアサーバー上で実行されます。

プラグインの要求フェーズでは、次の API が順次呼び出されます。

表 6-5 プラグインの要求フェーズ API 呼び出しの順番

| 段階              | API 呼び出し                                  | サンプルのプラグインの参照                      |
|-----------------|-------------------------------------------|------------------------------------|
| プラグインの初期化       | <code>「aapi_pgn_init」</code>              | aapi_pgn_init                      |
| PSF によるプラグインの要求 | <code>「aapi_pgn_claim」</code>             | samplepgn_claim_v1                 |
| 接続の設定           | <code>「aapi_open_server」</code>           | samplepgn_open_server_v1           |
| プラグインのアンロード     | <b>1</b> <code>「aapi_close_server」</code> | <b>1</b> samplepgn_close_server_v1 |
|                 | <b>2</b> <code>「aapi_terminate」</code>    | <b>2</b> samplepgn_terminate_v1    |

### NetBackup のプラグインの初期化フェーズでのログ情報の追加

既存のエラーとは別に、関連するデバッグログファイルにカスタムログを作成します。

p.92 の「[プラグインのログの有効化](#)」を参照してください。

### 関連項目

p.59 の「[開発中のプラグインの検証およびテスト](#)」を参照してください。

p.36 の「[サンプルプラグインについて](#)」を参照してください。

## プラグインのアンロードフェーズについて

検出、バックアップ、またはリストア処理の終了を示すために、プラグインがアンロードされます。

必要に応じて、プラグインをアンロードする前に、プラグインまたは作業負荷に固有のクリーンアップ操作を追加できます。

## 関連するプロセスと API

bpbrm プロセスは、メディアサーバー上で実行されます。

プラグインのアンロードフェーズでは、次の API が順次呼び出されます。

表 6-6 プラグインのアンロードフェーズ API 呼び出しの順番

| フェーズ        | API 呼び出し              | サンプルのプラグインの参照               |
|-------------|-----------------------|-----------------------------|
| プラグインのアンロード | 1 「aapi_close_server」 | 1 samplepgn_close_server_v1 |
|             | 2 「aapi_terminate」    | 2 samplepgn_terminate_v1    |

## NetBackup のプラグインの初期化フェーズでのログ情報の追加

既存のエラーとは別に、関連するデバッグログファイルにカスタムログを作成します。

p.92 の「[プラグインのログの有効化](#)」を参照してください。

## 関連項目

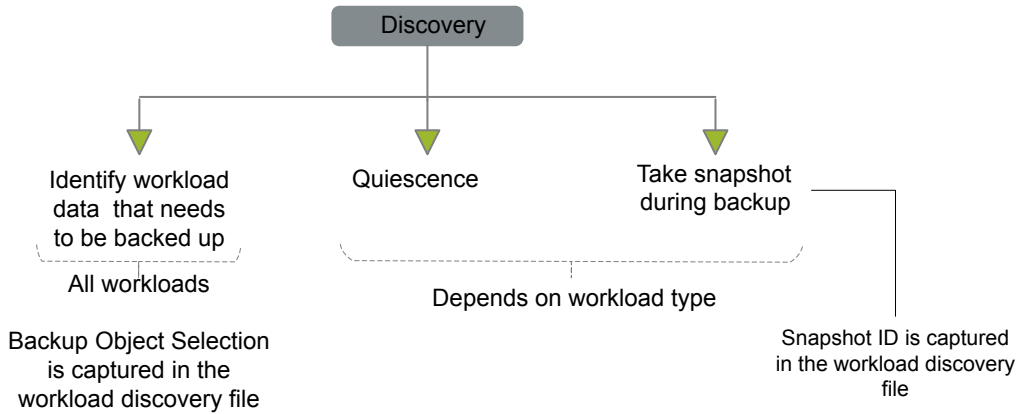
p.59 の「[開発中のプラグインの検証およびテスト](#)」を参照してください。

p.36 の「[サンプルプラグインについて](#)」を参照してください。

# 作業負荷検出フェーズについて

次の図は、検出フェーズ中に行われるさまざまなタスクを示しています。

図 6-1 作業負荷の検出フェーズ



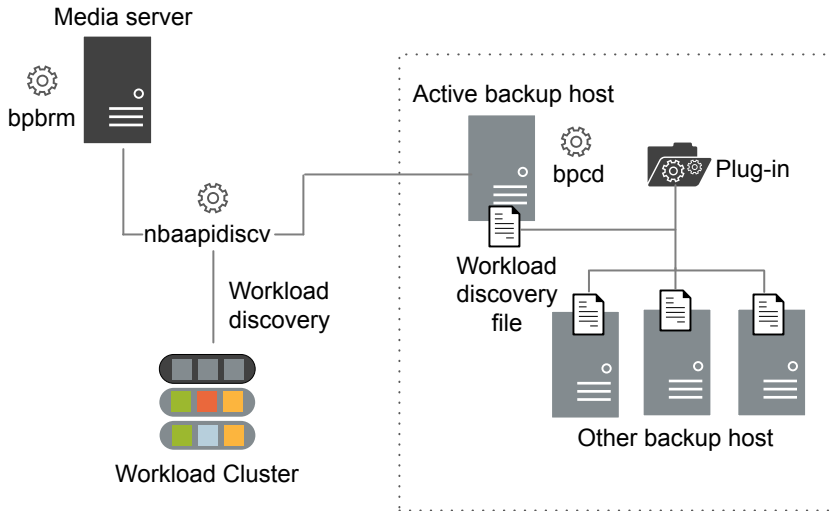
検出フェーズは、次のタスクで構成されます。

- (すべての作業負荷向け) アプリケーション固有の検出を実行し、各バックアップホストについて、バックアップする必要があるデータの詳細を確認します。
- (オプション) アプリケーションのバックアップに一貫性を保つために、アプリケーションデータを静止します。
- (作業負荷の種類に応じて) クラッシュ整合性のあるスナップショットをバックアップ中に作成します。

## 作業負荷の検出フェーズのプロセス

次の図に検出フェーズのプロセスを示します。

図 6-2 作業負荷の検出フェーズのプロセス



1. nbaapidiscv は、機能しているバックアップホストでプラグイン定義の検出 API を呼び出し、作業負荷アプリケーションの検出を実行します。

検出 API は、作業負荷ファイルの検出情報を、各バックアップホストに事前定義済みの形式で配置します。

2. プラグインは、複数のバックアップホストを使用してバックアップできるように、バックアップデータをどのように分割するかを決定します。

複数のバックアップホストをポリシーに構成すると、検出 API は、スナップショットとバックアップの関連情報を含めて、各バックアップホスト用に複数の作業負荷ファイルを作成します。この場合、複数のジョブが各バックアップホストで独立して実行され、各ジョブは、作業負荷ファイルで指定されたデータのみをバックアップします。

## プラグインに送信される情報

NetBackup Parallel Streaming Framework は、次の情報をプラグインに送信します。

|           |                                                     |
|-----------|-----------------------------------------------------|
| バックアップ対象  | 構成済みポリシーのバックアップ対象です。                                |
| バックアップホスト | 構成済みポリシーの利用可能なバックアップホストで、並列バックアップの操作を実行するために使用されます。 |

|                    |                                                                                                                                                                                                                                                                                                         |
|--------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 作業負荷出力ディレクトリ       | プラグインが作業負荷ファイルを作成する、作業負荷出力ディレクトリです。出力ディレクトリには、各バックアップホストの作業負荷ファイルが、 <code>backuphost.workloadfilesuffix</code> の形式で含まれています。                                                                                                                                                                             |
| ワークロードファイル名のサフィックス | 作業負荷ファイル名に付加するサフィックスです。ファイルを一意に識別できるようにするために、バックアップホスト用に作成される各作業負荷ファイル名に、このサフィックスを付加する必要があります。                                                                                                                                                                                                          |
| バックアップスケジュール形式     | バックアップスケジュールの種類で、完全、差分増分、累積増分があります。                                                                                                                                                                                                                                                                     |
| スナップショット ID        | <p>作業負荷データ用に作成されるスナップショットを識別するための、一意のスナップショット ID です。</p> <p><b>NetBackup Parallel Streaming Framework</b> では、バックアップジョブがトリガされたときのタイムスタンプをスナップショット ID として使用します。同じスナップショット ID は、作業負荷ファイル内と、後処理中に使用されます。</p> <p>必要に応じて、プラグイン開発者は、スナップショット ID をカスタマイズできます。<a href="#">p.47 の「作業負荷ファイルについて」</a>を参照してください。</p> |

## 関連するプロセスと API

作業負荷検出フェーズでは、次の **NetBackup** プロセスが順番に実行されます。

1. `bpbrm` プロセスは、メディアサーバー上で実行されます。
  2. `bpacd` プロセスは、機能しているバックアップホスト上で実行されます。
  3. `nbaapidiscv` は、機能しているバックアップホストを使用して検出を実行します。
- 検出フェーズでは、次の **API** が順番に呼び出されます。

**表 6-7**                      検出フェーズと API 呼び出しの順番

| フェーズ            | API 呼び出し                                  | サンプルのプラグインの参照                                   |
|-----------------|-------------------------------------------|-------------------------------------------------|
| プラグインの初期化       | <code>[aapi_pgn_init]</code>              | <code>aapi_pgn_init</code>                      |
| PSF によるプラグインの要求 | <code>[aapi_pgn_claim]</code>             | <code>samplepgn_claim_v1</code>                 |
| 接続の設定           | <code>[aapi_open_server]</code>           | <code>samplepgn_open_server_v1</code>           |
| 検出フェーズ          | <code>[aapi_discovery]</code>             | <code>samplepgn_discovery_v1</code>             |
| プラグインのアンロード     | <b>1</b> <code>[aapi_close_server]</code> | <b>1</b> <code>samplepgn_close_server_v1</code> |
|                 | <b>2</b> <code>[aapi_terminate]</code>    | <b>2</b> <code>samplepgn_terminate_v1</code>    |

## NetBackup のプラグインの初期化フェーズでのログ情報の追加

既存のエラーとは別に、関連するデバッグログファイルにカスタムログを作成します。

p.92 の「[プラグインのログの有効化](#)」を参照してください。

## 開発中のプラグインの検証およびテスト

開発中のプラグインをテストする際に、それぞれの開発フェーズのログを確認します。プラグイン全体を構築するまでは、ジョブが失敗する可能性があります。特定のフェーズの成功ログを確認する必要があります。

たとえば、検出フェーズの開発を完了した場合、バックアップは失敗する可能性があります。検出は成功するはずです。

p.59 の「[開発中のプラグインの検証およびテスト](#)」を参照してください。

## 関連項目

p.36 の「[サンプルプラグインについて](#)」を参照してください。

p.47 の「[作業負荷ファイルについて](#)」を参照してください。

## 作業負荷ファイルについて

プラグインは、検出プロセスの後、バックアップホストごとに作業負荷ファイルを作成します。各バックアップホストの作業負荷ファイルには、関連付けられているバックアップジョブの特定時点の情報を含める必要があります。新しい作業負荷ファイルをバックアップごとに作成する必要があります。作業負荷ファイルでは、スナップショット ID (スナップショットが有効なバックアップの場合) と、バックアップするデータを表すアプリケーションオブジェクト階層が使用されます。

## 作業負荷ファイルの構造

作業負荷ファイルはバイナリファイル形式で、構造は事前に定義されています。ただし、アプリケーションオブジェクト階層は、作業負荷に固有です。作業負荷ファイルの構造を次に示します。

```
snapshot_name_length (uint64) snapshot_name
  objects_path-1_length (uint 64) object_path-1
  objects_path-2_length (uint 64) object_path-2

  objects_path-n_length (uint 64) object_path-n
```

作業負荷ファイルの内容の例を次に示します。

```
50 snapshot_name=snapshot1_06-05-2015_18:08:25.048027
4 /vm1 (object path is maintained in the catalog)
11 /vm1/vdisk1 (object path is maintained in the catalog)
```

```
11 /vm1/vdisk2          (object path is maintained in the catalog)
4 /vm2                  (object path is maintained in the catalog)
11 /vm2/vdisk1          (object path is maintained in the catalog)
```

## 注意事項

- スナップショットが有効なバックアップの場合:
  - 作業負荷ファイルでは、アプリケーションオブジェクト階層の前に、スナップショット ID が示されています。
  - バックアップ処理は、有効なオブジェクトパスを参照するために、スナップショット ID (snapname) を使用してオブジェクトパス (realpath) を実際のパス (snappath) に解決します。このために、バックアップ処理はオブジェクトごとに「[aapi\\_get\\_snappath\\_from\\_realpath](#)」API を呼び出します。  
たとえば、スナップショット ID が `snapshot1_06-05-2018_18:08:25.048027` の `/vm1/vdisk1` オブジェクトを NetBackup がバックアップする場合、バックアップ処理では次の操作が実行されます。
    - スナップショット名 `snapshot1_06-05-2018_18:08:25.048027` と実際のパス `/vm1/vdisk1` を使用して、`aapi_get_snappath_from_realpath` API を呼び出して、実際のパス (snappath) を取得します。
    - 実際のパス `/snapshot1_06-05-2018_18:08:25.048027/vm2/vdisk1` を取得します。これは、ディスクオブジェクトを開いて読み取るために使用できます。
    - 実際のパス (snappath) `/snapshot1_06-05-2018_18:08:25.048027/vm2/vdisk1` を使用して、`aapi_open_object` を呼び出してディスクオブジェクトを開き、有効なオブジェクトパス `/vm1/vdisk1` を参照します。
  - 次に、バックアップ処理は実際のパスを使用して、「[aapi\\_get\\_object\\_prop\\_byname](#)」API によるオブジェクトプロパティ情報のクエリーを実行します。バックアップ処理では、マスターサーバー上の NetBackup カタログにオブジェクトのプロパティ情報が格納されます。カタログに格納されている情報は、リストア時に使用されます。
- スナップショット作成がプラグインでサポートされない場合、バックアップ処理は、作業負荷ファイルに示されているオブジェクトパスを実際のパスとして扱い、それらを使用して「[aapi\\_get\\_object\\_prop\\_byname](#)」API によるオブジェクトプロパティ情報を問い合わせます。
- 作業負荷ファイルは、次の場所に作成されます。  
`/install_path/tmp/masterserver/mediaserver/backuphost/policyname/apptype`

---

**メモ:** 作業負荷ファイルの安全を確保するには、作業負荷ファイルを作成するときに、ルートのみで ACL を設定する必要があります。

---



## カスタムスナップショット ID の使用

必要に応じて、プラグイン開発者は、スナップショット ID をカスタマイズできます。カスタマイズしたスナップショット ID を使用する場合、この ID は、NetBackup が作業負荷ファイルで生成したスナップショット ID に置き換えられます。ただし、後処理中は、NetBackup が提供した元のスナップショット ID のみが使用されます。そのため、開発者は、元のスナップショット ID とカスタマイズされたスナップショット ID のマッピングを管理する必要があります。

また、カスタムの ID を使用するときと、NetBackup Parallel Streaming Framework が実際のパスのスナップショットパスを要求するときは必ず、NetBackup は元のスナップショット ID を使用します。そのため、開発者は、元のスナップショット ID とカスタマイズされたスナップショット ID のマッピングを管理する必要があります。

また、NetBackup Parallel Streaming Framework が実際のパスのスナップショットパスを要求した場合、NetBackup は常に元のスナップショット ID を使用します。そのため、カスタム ID を使用する場合は、開発者が元のスナップショット ID とカスタマイズされたスナップショット ID のマッピングを管理し、対応する有効なスナップパスを Parallel Streaming Framework に返す必要があります。

## データバックアップフェーズについて

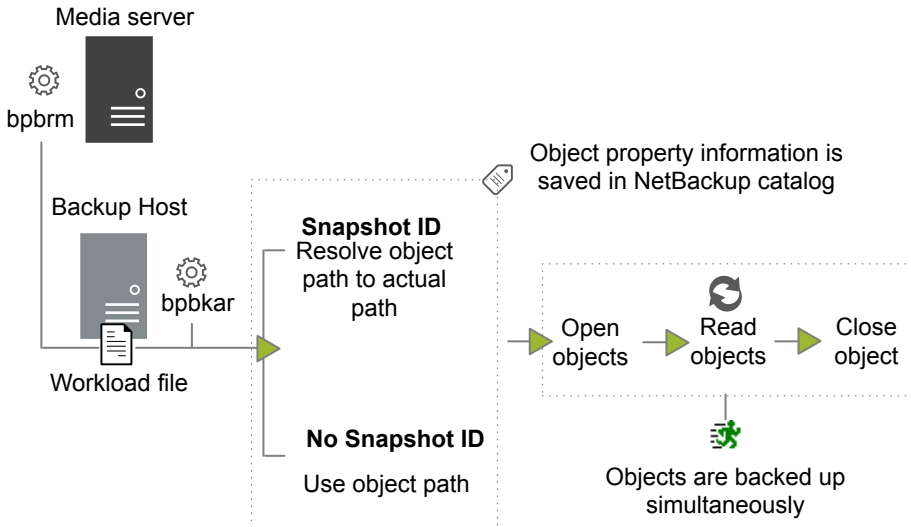
検出ジョブが正常に完了したら、メディアサーバー上の bpbrm プロセスが対応するバックアップホストに作業負荷ファイルを配布し、バックアップホスト上でバックアップタスクを開始します。バックアップジョブは、検出ジョブとは異なり、BigData ポリシーに構成されているすべてのバックアップホストで実行されます。p.12 の「[Parallel Streaming Framework を使用したバックアップ処理](#)」を参照してください。

バックアップ処理が開始されると、メディアサーバー上で bpbrm プロセスが実行されて、すべてのバックアップホストでバックアップフェーズ (bpbkar) がアクティブ化されます。

### バックアップフェーズのプロセス

次の図に、バックアップフェーズのプロセスを示します。

図 6-3 バックアップフェーズのプロセス



1. bpbkar プロセスは、作業負荷検出フェーズ中に作成された作業負荷ファイルを読み込みます。作業負荷ファイルには、スナップショットの識別子 (スナップショットが有効なバックアップの場合) とアプリケーションオブジェクト階層が含まれています。p.47 の「[作業負荷ファイルについて](#)」を参照してください。
2. 作業負荷によっては、次の手順のいずれかを実行します。
  - スナップショットが有効なバックアップの場合:
    - 作業負荷ファイルでは、アプリケーションオブジェクト階層の前に、スナップショット ID が示されています。
    - バックアップ処理は、有効なオブジェクトパスを参照するために、スナップショット ID (snapname) を使用してオブジェクトパス (realpath) を実際のパス (snappath) に解決します。このために、バックアップ処理はオブジェクトごとに「[aapi\\_get\\_snappath\\_from\\_realpath](#)」API を呼び出します。
    - 次に、バックアップ処理は実際のパスを使用して、「[aapi\\_get\\_object\\_prop\\_byname](#)」API によるオブジェクトプロパティ情報のクエリーを実行します。バックアップ処理では、マスターサーバー上の NetBackup カタログにオブジェクトのプロパティ情報が格納されます。カタログに格納されている情報は、リストア時に使用されます。

- スナップショットが有効なバックアップの場合、作業負荷ファイルにはスナップショット識別子が含まれます。これは、アプリケーションオブジェクト階層の前に示されます。

バックアップ処理は、有効なオブジェクトパスを参照するために、スナップショット識別子 (snapname) を使用してオブジェクトパス (realpath) を実際のパス (snappath) に解決します。このために、バックアップ処理はオブジェクトごとに「aapi\_get\_snappath\_from\_realpath」API を呼び出します。次に、バックアップ処理は実際のパスを使用して、プラグインが実装する

aapi\_get\_object\_prop\_byname API によるオブジェクトプロパティ情報のクエリーを実行します。バックアップ処理では、さまざまな目的で、マスターサーバー上の NetBackup カタログにオブジェクトのプロパティ情報が格納されます。

3. バックアップ処理は「aapi\_open\_object」API を呼び出して、実際のパスを使用してオブジェクトを開きます。
4. 成功すると、「aapi\_read\_object」API が繰り返し呼び出されて、オブジェクトが表示データが読み込まれます。
5. 読み込まれたデータは、データ移動用にメディアサーバーに送信されます。
6. 読み込みが完了すると、「aapi\_close\_object」API が呼び出されてオブジェクトが閉じられます。

## 関連するプロセスと API

バックアップフェーズでは、次の NetBackup プロセスが順番に実行されます。

1. bpbrm プロセスは、メディアサーバー上で実行されます。
2. bpbkar は、すべてのバックアップホストでバックアップを実行します。

バックアップフェーズでは、次の API が順番に呼び出されます。

表 6-8 バックアップフェーズの API 呼び出しの順番

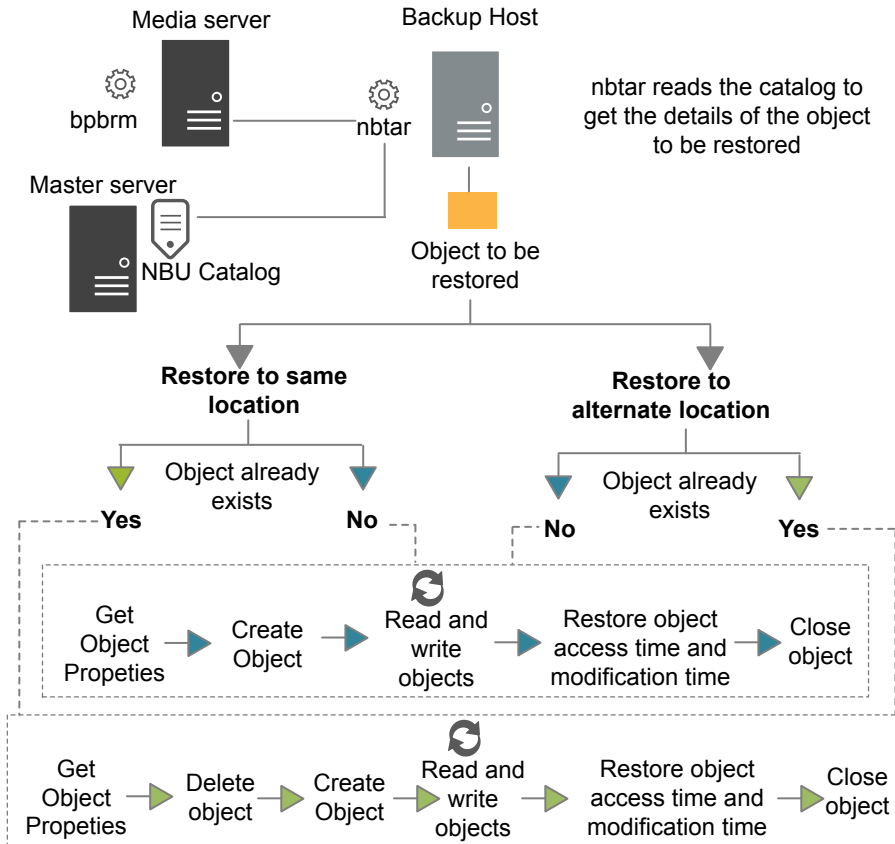
| フェーズ            | API 呼び出し           | サンプル API リファレンス          |
|-----------------|--------------------|--------------------------|
| プラグインの初期化       | 「aapi_pgn_init」    | aapi_pgn_init            |
| PSF によるプラグインの要求 | 「aapi_pgn_claim」   | samplepgn_claim_v1       |
| 接続の設定           | 「aapi_open_server」 | samplepgn_open_server_v1 |



## リストア処理

リストアタスクは、次の図に示すように nbtar プロセス主導で進められます。

図 6-4 リストアフェーズの処理



1. 代替リストアの場合、オブジェクトのリストアパスは、バックアップされたオブジェクトパスから代替リストアパスに変更されます。

たとえば、バックアップオブジェクトパスが

/obj-container1/obj-container2/file.txt の場合、これは

/mnt/vol-1/obj-container1/obj-container2/file.txt に変わります (代替の場所は mnt/vol-1)。

2. 「`aapi_get_object_prop_byname`」API は、オブジェクトプロパティを取得するために、オブジェクトリストアパスを使用して呼び出されます。

たとえば、`aapi_get_object_prop_byname` API

は、`/mnt/vol-1/obj-container1/obj-container2/file.txt` のメタプロパティを提供します。

3. 4. オブジェクトリストアパスがすでに存在し、次の条件が該当する場合。
  - 上書きオプションが `false` に設定されている場合、オブジェクトのリストアはスキップされます。
  - 上書きオプションが `true` に設定されている場合、既存のオブジェクトを削除するために「`aapi_delete_object`」API が呼び出されます。
4. 「`aapi_create_object`」API の実装を使用して、新しいオブジェクトが作成されます。

名前、サイズ、アクセス権、`u/mtime`、所有者などのオブジェクト情報は、オブジェクトプロパティの設定に使用する `aapi_create_object` の呼び出しに渡されます。

`aapi_create_object` API は、リストアパスのすべてのオブジェクト (最初のオブジェクトからリーフオブジェクトまで) に対して呼び出されます。したがって、オブジェクトコンテナが存在しない場合は、これらも作成されます。

たとえば、`/mnt/vol-1/obj-container1/obj-container2/file.txt` をリストアする場合、`aapi_create_object` API は階層ごとに 1 回ずつで合計 5 回呼び出されます。

- `/mnt` - オブジェクトコンテナ
- `/mnt/vol-1` - オブジェクトコンテナ
- `/mnt/vol-1/obj-container1` - オブジェクトコンテナ
- `/mnt/vol-1/obj-container1/obj-container2` - オブジェクトコンテナ
- `/mnt/vol-1/obj-container1/obj-container2/file.txt` - データコンテナ

5. オブジェクトは、バックアップイメージからバックアップ済みコンテンツを読み込み、「`aapi_write_object`」API を呼び出すことでリストアされます。

このプロセスは、オブジェクト全体が読み込まれるまで、順番に繰り返し実行されます。

6. 「`aapi_set_object_utimes`」API が呼び出されて、オブジェクトのアクセス時刻と修正時刻がリストアされます。
7. すべてのオブジェクトがリストアされた後、「`aapi_set_object_utimes`」API が再び呼び出されて、オブジェクトコンテナのアクセス時刻と修正時刻がリストアされます。

8. 「aapi\_close\_object」API を呼び出してオブジェクト関連のクリーンアップを実行して、データオブジェクトが閉じられます。

## 関連するプロセスと API

リストアフェーズでは、次の NetBackup プロセスが順番に実行されます。

1. bpbm プロセスは、メディアサーバー上で実行されます。
2. nbstar は、すべてのバックアップホストでバックアップを実行します。

リストアフェーズでは、次の API が順番に呼び出されます。

表 6-9 リストアフェーズの API 呼び出しの順番

| フェーズ            | API 呼び出し                        | サンプルのプラグインの参照                         |
|-----------------|---------------------------------|---------------------------------------|
| プラグインの初期化       | 「aapi_pgn_init」                 | aapi_pgn_init                         |
| PSF によるプラグインの要求 | 「aapi_pgn_claim」                | samplepgn_claim_v1                    |
| 接続の設定           | 「aapi_open_server」              | samplepgn_open_server_v1              |
| リストアフェーズ        | 1 「aapi_get_object_prop_byname」 | 1 samplepgn_get_object_prop_byname_v1 |
|                 | 2 「aapi_delete_object」          | 2 samplepgn_delete_object_v1          |
|                 | 3 「aapi_create_object」          | 3 samplepgn_create_object_v1          |
|                 | 4 「aapi_set_object_utimes」      | 4 samplepgn_set_object_utimes_v1      |
|                 | 5 「aapi_write_object」           | 5 samplepgn_write_object_v1           |
|                 | 6 「aapi_close_object」           | 6 samplepgn_close_object_v1           |
| プラグインのアンロード     | 1 「aapi_close_server」           | 1 samplepgn_close_server_v1           |
|                 | 2 「aapi_terminate」              | 2 samplepgn_terminate_v1              |

## NetBackup のプラグインの初期化フェーズでのログ情報の追加

既存のエラーとは別に、関連するデバッグログファイルにカスタムログを作成します。

p.92 の「プラグインのログの有効化」を参照してください。

## 開発中のプラグインの検証およびテスト

開発中のプラグインをテストする際に、それぞれの開発フェーズのログを確認します。プラグイン全体を構築するまでは、ジョブが失敗する可能性があります。特定のフェーズの成功ログを確認する必要があります。

たとえば、検出フェーズの開発を完了した場合、バックアップは失敗する可能性があります。検出は成功するはずです。

p.59 の「[開発中のプラグインの検証およびテスト](#)」を参照してください。

## 関連項目

p.13 の「[Parallel Streaming Framework を使用したリストア処理](#)」を参照してください。

p.36 の「[サンプルプラグインについて](#)」を参照してください。

# リカバリフェーズについて

作業負荷プラグインの効率的なリカバリ戦略を、バックアップフェーズから計画する必要があります。リカバリフェーズは、保護対象となる作業負荷によって異なります。たとえば、仮想マシンのリカバリと比較すると、データベースには異なるリカバリ手順が必要です。

次の手順では、データリカバリ処理の概要を示します。

### 1. 「作業負荷ファイルへのリカバリメタデータの追加」

このサブフェーズでは、リカバリ情報をキャプチャするために、リカバリメタデータが作業負荷ファイルに追加されます。バックアップ処理では、必要な情報をキャプチャし、**NetBackup** カタログを更新します。

### 2. 「作業負荷データのリカバリ」

このサブフェーズでは、リストア処理が **NetBackup** カタログからリカバリのメタデータを読み取り、リカバリ情報に従って作業負荷データをリカバリします。

## リカバリフェーズの考慮事項

- 1 つの親オブジェクトのすべてのメタデータとリカバリファイルを、1 つのバックアップホストを使用してバックアップする必要があります。
- リストア処理中にリカバリメタデータを変更することはできません。

## リカバリフェーズを計画するための **NetBackup PSF API**

リカバリフェーズでは、次の **API** を使用します。

表 6-10 作業負荷リカバリ用の **API**

| API                                     | 説明                                                                                     |
|-----------------------------------------|----------------------------------------------------------------------------------------|
| <code>「aapi_restore_preprocess」</code>  | サービスのシャットダウンやデータベースのマウント解除などのリカバリ前のタスクを実行するためのリカバリフェーズで、フレームワークはこの <b>API</b> を呼び出します。 |
| <code>「aapi_restore_postprocess」</code> | 作業負荷データのリカバリを開始するために、フレームワークはこの <b>API</b> を呼び出します。                                    |
| <code>「aapi_set_object_utimes」</code>   | 完全なリカバリ情報を収集して、後処理でできるようにするために、フレームワークはこの <b>API</b> を呼び出します。                          |



## 作業負荷ファイルへのリカバリメタデータの追加

バックアップオブジェクトの検出中に、プラグインは作業負荷ファイルを使用してリカバリメタデータを格納できます。リカバリメタデータは、データリカバリフェーズで取得および使用されます。

メタデータには、バックアップオブジェクトの階層、バックアップ対象の親オブジェクト、すべてのファイルの場所とリスト、および作業負荷データのリカバリに必要な情報が含まれます。ファイルには、リカバリ処理で必要になるすべての物理ファイル、追加のメタデータファイル、またはマニフェストファイルを含めることができます。

1. 追加のメタデータを収集するために、作業負荷ファイルにブレースホルダを追加します。作業負荷ファイルの追加のメタデータは新しいノードにでき、親オブジェクトと追加のリカバリ項目に関する情報が含まれます。バックアップオブジェクトは、親オブジェクトとメタデータの下に表示できます。

p.47 の「[作業負荷ファイルについて](#)」を参照してください。

たとえば、次に示す作業負荷ファイルのスニペットでは、作業負荷のリカバリのために、作業負荷に実際のバックアップ対象 `backup_selection`、スナップショットディレクトリ `snapshot_dir`、メタファイル `manifest_file_name`、およびスナップショット情報 `snapshot_file_name` が必要です。

```
/backup_selection/  
/backup_selection/recovery_metadata/  
/backup_selection/recovery_metadata/checkpoint  
/backup_selection/recovery_metadata/checkpoint/checkpoint_file_name  
/backup_selection/recovery_metadata/snapshot_dir  
/backup_selection/recovery_metadata/snapshot_dir/manifest_file_name  
/backup_selection/recovery_metadata/snapshot_dir/snapshot_file_name  
/backup_selection/backup_data/  
/backup_selection/backup_data/logs  
/backup_selection/backup_data/logs/log_file_name  
/backup_selection/backup_data/data  
/backup_selection/backup_data/data/database_file_name
```

---

**メモ:** この例の作業負荷ファイルには、バックアップ用のデータが含まれています。このバックアップデータは、`recovery_metadata` と `backup_data` の各セクションに分離されます。`recovery_metadata` セクションには、成功したリカバリの情報が含まれ、`backup_data` セクションには、実際の作業負荷データが含まれています。

---

2. リカバリメタデータのセクションが追加されると、作業負荷プラグインでは、作業負荷ファイルをスキャンし、リカバリ関連の必要な情報をキャプチャし、バックアップイメージを準備してバックアップを完了します。

p.49 の「データバックアップフェーズについて」を参照してください。

## 作業負荷データのリカバリ

作業負荷データのリカバリは、作業負荷ファイルでキャプチャされたリカバリメタデータに基づいたリストアフェーズと、データのリカバリで構成されます。

1. リストアフェーズでは、フレームワークが `aapi_set_object_utimes` API を呼び出すと、リストア処理がリカバリメタデータを読み取り、その後データリカバリに必要なすべての情報を収集します。リカバリ情報とリカバリ可能な項目がキャプチャされ、ファイルまたはメモリ内に保持されます。

リストアフェーズでは、フレームワークが `aapi_set_object_utimes` API を呼び出すと、リカバリ可能な項目がキャプチャされて、ファイルまたはメモリ内に保持されます。その後、API は、データリカバリに必要なすべての情報を収集します。

p.52 の「リストアフェーズについて」を参照してください。

たとえば、リカバリメタデータの追加の手順 1 で指定された作業負荷ファイルのスニペットで、`aapi_set_object_utimes` の呼び出しはまず

`/backup_selection/backup_data/data/database_file_name` の情報を読み取ります。API は、次に `recovery_metadata` エントリを検索し、リカバリ可能な対象の `/backup_selection/recovery_metadata/`、スナップショットディレクトリの `snapshot_dir`、およびその他の情報を検索します。

2. フレームワークは `aapi_restore_postprocess` API を呼び出し、`aapi_set_object_utimes` の呼び出し後に、リストア済みオブジェクトのリカバリを開始します。リカバリ手順は、保護する作業負荷に基づいています。たとえば、データベースをリカバリしてテーブルとしてマウントしたり、データベースサービスをアクティブにしたりできます。仮想マシンの場合は、リストアされたマシンを Hypervisor または VMware に登録するように指示できます。

## API の実行時間の管理

API のいずれかで、NetBackup で定義されたタイムアウト間隔よりも長い時間が実行に必要な場合、API は、NetBackup に `keep alive` シグナルを送信することになっています。

実装されているすべての API が非ブロッキングであり、「クライアント読み込み」のタイムアウトが切れる前に戻ることを確認する必要があります。このタイムアウトは、デフォルトでは 300 秒に設定されています。タイムアウト値は、NetBackup 管理コンソールのホストのプロパティを更新することで変更できます。

(これはオプションの要件です。)ただし、より長い (タイムアウトより長い) 処理時間を必要とするすべてのプラグイン API は、メディアサーバー上の NetBackup プロセスに `keep alive` シグナルを送信する必要があります。NetBackup プロセスに `keep alive` シグナル

を送信するには、「`aapi_open_server`」API 呼び出しの際に提供される、`aapi_session_def_t` 構造の `sd_heartbeat_callback` メソッドを使用します。

「aapi\_open\_server」で利用可能な、次のセッション変数を使用できます。

```
aapi_session_def_t *sd = getGlobalSessionDef();
```

```
aapi_heartbeat_callback_t heartbeat_callback =
sd->sd_heartbeat_callback;
```

```
int rc = AAPI EOK;
```

```

if (keepalive_callback)
{
    rc = keepalive_callback("api_name");
    if (AAPI_EOK != rc)
    {
        // log error
        return rc;
    }
}

```

## 開発中のプラグインの検証およびテスト

各フェーズの開発をテストおよび検証するには

1. **BigData** ポリシーを作成します。
2. バックアップホスト上の `usr/opencv/lib` ディレクトリに `.so` ファイルをコピーします。
3. バックアップまたはリストアジョブを実行します。
4. 該当する開発フェーズのログを確認します。プラグイン全体を構築するまでは、ジョブが失敗する可能性があります。特定のフェーズの成功ログを確認する必要があります。

たとえば、検出フェーズの開発を完了した場合、バックアップは失敗する可能性があります。検出は成功するはずです。

## 関連項目

p.59 の「作業負荷プラグインのパッケージ化」を参照してください。

## 作業負荷プラグインのパッケージ化

プラグインのパッケージは、tar.gz ファイルまたは rpm ファイルにできます。

また、安全に配備するために、プラグインに署名することをお勧めします。

### プラグインをコンパイルするには

- 1 次のコマンドを実行します。

```
make libaapipgnworkloadname.so
```

- 2 次のファイルを含む **tar.gz** パッケージが作成されます。

- .tar ファイル
- install ファイル
- License ファイル

- 3 プラグインをダウンロード場所にアップロードします。

### 注意事項

プラグインをパッケージ化するときは、次の点を確認します。

- プラグインのバイナリは、ルートのみアクセス権を使用して配備する必要があります。
- NetBackup インストールパスにプラグインを配備するには、プラグインを事前定義済みの `/usr/opensv/lib/psf-plugins/third_party/plugin_name` パスに配備する必要があります。ここで、**plug-in name** をアプリケーションの種類と同じにすることをお勧めします。

# ワークロードプラグインの配備

この章では以下の項目について説明しています。

- [NetBackup File Checksum Validator](#) ツールを使用したプラグインの検証
- [バックアップホストへの作業負荷プラグインのインストール](#)
- 作業負荷向けの [NetBackup](#) の構成
- 作業負荷のバックアップとリストアの実行

## NetBackup File Checksum Validator ツールを使用したプラグインの検証

プラグインのコンシューマは、ダウンロードしたプラグインを検証する必要があります。  
NetBackup File Checksum Validator (nbfcv) コマンドラインツールは、プラグインのダウンロード後に、ファイルの sha256 チェックサムを計算して検証します。NetBackup File Checksum Validator (nbfcv) コマンドラインツールは、Veritas SORT のサイトから入手できます。

このユーティリティを使用すると、次のことが可能です。

- コンシューマは、すべての重要なファイルのチェックサムを計算できます。
- システムが危殆化していることが疑われる場合に、コンシューマは、どのファイルが変更されているかを判断するために検証を実行できます。

### 前提条件

- 次の場所から最新のチェックサムをダウンロードします。  
[https://sort.veritas.com/data/nbu/psf/nbu\\_psf\\_partnerplugins\\_info.json](https://sort.veritas.com/data/nbu/psf/nbu_psf_partnerplugins_info.json)

- ベストプラクティスとして、常に認定されたバージョンのプラグインを使用します。

## 使用方法

```
./nbfcv -input_path input -checksum_file json_file [-print_checksum]
[-verbose [verbose]] [-help]
```

## オプション

-input\_path

このオプションを使用して、ディレクトリに含まれるファイルのチェックサムを計算して検証するために、入力ファイルまたはディレクトリパスを指定します。

-checksum\_file オプションを指定してこのオプションを使用します。

-checksum\_file オプションの値を指定しない場合、このツールはベリタスサイトからこのファイルをダウンロードします。

-checksum\_file

このオプションを使用して、NetBackup の認定ファイルのチェックサムデータを提供する、利用可能な JSON 形式のファイルを提供します。

---

**メモ:** チェックサムファイルの内容は変更しないでください。

---

-print\_checksum

このオプションを -input\_path オプションと一緒に使用して、入力ディレクトリのファイルのチェックサム情報を出力します。

-verbose

このオプションを使用して、診断目的の詳細レベルを設定します。

サポートされる値:

- 1: 最小限
- 2: 基本 (デフォルトの詳細レベル)
- 3: 詳細

## 例

- 例 1: 入力ディレクトリのファイルのチェックサム検証

```
./nbfcv -input_path /usr/opensv/lib/psf-plugins/third-party
./nbfcv -input_path /usr/opensv/lib/psf-plugins/third-party
-checksum_file /usr/opensv/nbu_psf_partnerplugins_info.json
```

- 例 2: ファイルのチェックサムの検証

- ```
./nbfcv -input_path  
/usr/opensv/lib/psf-plugins/third-party/DB_plugin.rpm  
./nbfcv -input_path  
/usr/opensv/lib/psf-plugins/third-party/DB_plugin.rpm -checksum_file  
/usr/opensv/nbu_psf_partnerplugins_info.json
```
- 例 3: 入力ディレクトリ (再帰) のファイルの sha256 チェックサム情報の出力  

```
./nbfcv -print_checksum -input_path  
/usr/opensv/lib/psf-plugins/third-party
```
  - 例 4: 詳細レベル 3 でのログの出力  

```
./nbfcv -input_path /usr/opensv/lib/psf-plugins/third-party  
-checksum_file /usr/opensv/nbu_psf_partnerplugins_info.json -verbose  
3
```

## バックアップホストへの作業負荷プラグインのインストール

バックアップホストとして使用するすべてのメディアサーバー、クライアント、またはマスターサーバー上に作業負荷プラグインをインストールする必要があります。

---

**メモ:** NetBackup でプラグインを使用するためには、アプリケーションとデータベースパックライセンスが必要です。

---

### バックアップホストに作業負荷プラグインをインストールするには

- 1 ルート権限でバックアップホストにログインします。
- 2 次のファイル形式が抽出されます。
- 3 プラグインのバイナリ (.so ファイル)  
は、`/usr/opensv/lib/psf-plugins/third-party/vendor_name/plugin_name`  
パスにインストールされます。ここで、**plug-in name** をアプリケーションの種類と同じ  
にすることをお勧めします。

パッケージの種類に応じて、関連するコマンドを使用してパッケージを抽出する必要があります。たとえば、**tar** パッケージを指定する場合は、次のコマンドを実行します。

1. プラグインを抽出します。次のコマンドを実行します。

```
tar -xvf package_name
```

2. インストーラを実行します。次のコマンドを実行します。

```
./install
```

## 作業負荷向けの NetBackup の構成

表 7-1 作業負荷向けの NetBackup の構成

| 作業                        | 参照先                                                                                                                                                                                                                          |
|---------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| バックアップホストの追加              | p.27 の「 <a href="#">バックアップホストの管理</a> 」を参照してください。<br>NetBackup クライアントをバックアップホストとして使用する場合、マスターサーバー上の NetBackup クライアントをホワイトリストに載せる必要があります。<br>p.29 の「 <a href="#">NetBackup マスターサーバー上の NetBackup クライアントのホワイトリスト</a> 」を参照してください。 |
| NetBackup でのクレデンシヤルの追加    | p.33 の「 <a href="#">NetBackup での作業負荷クレデンシヤルの追加</a> 」を参照してください。                                                                                                                                                               |
| プラグイン用の NetBackup ポリシーの構成 | p.29 の「 <a href="#">BigData バックアップポリシーの作成</a> 」を参照してください。                                                                                                                                                                    |

## 作業負荷のバックアップとリストアの実行

NetBackup のバックアップ、アーカイブおよびリストアコンソールを使用して、バックアップとリストア操作を管理します。

### クラスタのバックアップについて

バックアップジョブはスケジュール設定して実行することもできれば、手動で実行することもできます。『[NetBackup 管理者ガイド Vol. 1](#)』を参照してください。

バックアップ処理の概要については、p.12 の「[Parallel Streaming Framework を使用したバックアップ処理](#)」を参照してください。

バックアッププロセスは、次のステージで構成されます。

1. 事前処理: 事前処理のステージでは、BigData ポリシーで構成した最初のバックアップホストが検出をトリガします。
2. データ転送: データ転送処理中には、バックアップホストごとに 1 つの子ジョブが作成されます。
3. 事後処理: 事後処理の一部として、NetBackup はスナップショットをクリーンアップします。

### 作業負荷クラスタのリストアについて

リストア処理の概要については、p.13 の「[Parallel Streaming Framework を使用したリストア処理](#)」を参照してください。



次のいずれかの方法を使用して、同じデータノードまたは代替データノード上のデータをリストアできます。

バックアップ、アーカイブおよびリストアインターフェース。『[NetBackup 管理者ガイド Vol. 1](#)』を参照してください。

bprestore コマンド。『[NetBackup コマンドリファレンスガイド](#)』を参照してください。

# NetBackup Parallel Streaming Framework API リファレンス

この章では以下の項目について説明しています。

- [aapi\\_close\\_object](#)
- [aapi\\_close\\_server](#)
- [aapi\\_create\\_object](#)
- [aapi\\_delete\\_object](#)
- [aapi\\_discovery](#)
- [aapi\\_flush](#)
- [aapi\\_get\\_object\\_prop](#)
- [aapi\\_get\\_object\\_prop\\_byname](#)
- [aapi\\_get\\_server\\_prop](#)
- [aapi\\_get\\_snappath\\_from\\_realpath](#)
- [aapi\\_open\\_object](#)
- [aapi\\_open\\_server](#)
- [aapi\\_pgn\\_claim](#)
- [aapi\\_pgn\\_init](#)
- [aapi\\_postprocess](#)

- [aapi\\_read\\_object](#)
- [aapi\\_set\\_object\\_utimes](#)
- [aapi\\_terminate](#)
- [aapi\\_write\\_object](#)
- [aapi\\_restore\\_preprocess](#)
- [aapi\\_restore\\_postprocess](#)
- [NetBackup Parallel Streaming Framework API の戻り値](#)

# aapi\_close\_object

## 概要

```
int
aapi_close_object(
    aapi_pgn_object_handle_t object_handle,
    int mode)
```

## 説明

NetBackup PSF は、この API を呼び出して、オブジェクトハンドルをプラグインに送信します。

プラグインがオブジェクトハンドルを受け取ると、プラグインはオブジェクトを閉じて、オブジェクトのプロパティをデフォルト状態にリセットする必要があります。

## パラメータ (Parameters)

表 8-1 API のパラメータ

| パラメータ         | 型         | 説明           |
|---------------|-----------|--------------|
| object_handle | 入力: const | オブジェクトハンドル   |
| mode          | 入力: int   | 現在使用されていません。 |

## 戻り値

表 8-2 戻り値

| 値        | 説明           |
|----------|--------------|
| AAPI_EOK | 成功 (Success) |

| 値         | 説明              |
|-----------|-----------------|
| AAPI_EIO  | 入力出力エラーです。      |
| AAPI_ESYS | システムコールに失敗しました。 |

すべての戻り値のリストは、p.90 の「[NetBackup Parallel Streaming Framework API の戻り値](#)」を参照してください。

## aapi\_close\_server

### 概要

```
int  
aapi_close_server(aapi_pgn_server_handle_t server_handle)
```

### 説明

PSF は、サーバーハンドルをクリアし、サーバーハンドルに関連付けられているすべてのリソースを解放するために、この API を呼び出します。

### パラメータ (Parameters)

表 8-3 API のパラメータ

| パラメータ         | 型         | 説明       |
|---------------|-----------|----------|
| server_handle | 入力: const | サーバーハンドル |

### 戻り値

表 8-4 戻り値

| 値        | 説明           |
|----------|--------------|
| AAPI_EOK | 成功 (Success) |

すべての戻り値のリストは、p.90 の「[NetBackup Parallel Streaming Framework API の戻り値](#)」を参照してください。

# aapi\_create\_object

## 概要

```
aapi_create_object(  
    const aapi_pgn_server_handle_t  server_handle,  
    const char                      *pathname,  
    int                             modes,  
    aapi_pgn_object_handle_t        *pobject_handle,  
    const aapi_object_info_v1_t     *objinfo)
```

## 説明

PSF は、リストア処理中にこの API を呼び出します。

プラグインは、指定された場所 (pathname) にオブジェクトを作成します。オブジェクトを作成すると、プラグインはオブジェクトのプロパティを設定し、オブジェクトハンドルを返します。

## パラメータ (Parameters)

表 8-5 API のパラメータ

| パラメータ          | 型              | 説明                                                                                                   |
|----------------|----------------|------------------------------------------------------------------------------------------------------|
| server_handle  | 入力: const      | サーバーハンドル                                                                                             |
| pathname       | 入力: const char | オブジェクトまたはエンティティをリストアするため、リストア操作で指定したパス。                                                              |
| modes          | 入力: int        | オブジェクトを作成するモード。                                                                                      |
| objinfo        | 入力: const      | オブジェクトが持つオブジェクトプロパティのポインタ。たとえば、オブジェクト形式、所有者、グループなどです。<br><br>このパラメータは、オブジェクトパスなどのカスタム値を追加するのに使用できます。 |
| pobject_handle | 出力             | 開かれたオブジェクトを参照するオブジェクトハンドルへのポインタ。                                                                     |

## 戻り値

表 8-6                    戻り値

| 値                 | 説明                             |
|-------------------|--------------------------------|
| AAPI_EOK          | 成功 (Success)                   |
| AAPI_EIO          | 入力出力エラーです。                     |
| AAPI_ESNAPDIRPATH | ディレクトリまたはパスを作成できないか、アクセスできません。 |
| AAPI_EEXIST       | 操作の完了に失敗しました。オブジェクトはすでに存在します。  |
| AAPI_ESYS         | システムコールに失敗しました。                |

すべての戻り値のリストは、p.90 の「[NetBackup Parallel Streaming Framework API の戻り値](#)」を参照してください。

# aapi\_delete\_object

## 概要

```
int
aapi_delete_object(
    aapi_pgn_server_handle_t server_handle,
    const char                *pathname,
    int                       recurse)
```

## 説明

NetBackup PSF は、リストア操作中にこの API を呼び出します。

この API は、削除するオブジェクト (pathname) を指定します。再帰パラメータは、子オブジェクトを再帰的に削除するかどうかを示します。

## パラメータ (Parameters)

表 8-7                    API のパラメータ

| パラメータ         | 型              | 説明        |
|---------------|----------------|-----------|
| server_handle | 入力: const      | サーバーハンドル  |
| pathname      | 入力: const char | オブジェクトのパス |

| パラメータ   | 型       | 説明                      |
|---------|---------|-------------------------|
| recurse | 入力: int | 子オブジェクトを再帰的に削除するかどうかを示す |

戻り値

表 8-8                    戻り値

| 値        | 説明           |
|----------|--------------|
| AAPI_EOK | 成功 (Success) |

すべての戻り値のリストは、p.90 の「[NetBackup Parallel Streaming Framework API の戻り値](#)」を参照してください。

aapi\_discovery

概要

```
int aapi_discovery(
    const aapi_pgn_server_handle_t server_handle,
    const char *selections,
    const char *backuphosts,
    const char *wkldoutdir,
    const char *wkldfilesuffix,
    const char *schedtype,
    const char *snapid)
```

説明

NetBackup PSF は、検出プロセスが開始されるとこの API を呼び出します。

この API は、バックアップ対象、バックアップホスト、作業負荷出力ファイルの場所と作業負荷ファイル名のサフィックス、バックアップスケジュール形式、スナップショット ID、およびサーバーハンドルをプラグインに送信します。

プラグインは、指定された場所にある各バックアップホストで、検出を実行し、スナップショットを作成し、作業負荷出力ファイルを生成します。

作業負荷ファイルは、バイナリファイル形式です。

パラメータ (Parameters)

表 8-9 API のパラメータ

| パラメータ           | 型              | 説明                                                                                                                                                                                                   |
|-----------------|----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| server_handle   | 入力: const      | サーバーハンドル                                                                                                                                                                                             |
| selections      | 入力: const char | 構成済みポリシーのバックアップ対象です。<br><br>複数のバックアップ対象は、スペースで区切ります。                                                                                                                                                 |
| wlkddoutdir     | 入力: const char | プラグインが作業負荷ファイルを作成する出力ディレクトリ。<br><br>作業負荷ファイル名の形式は、backuphost.wlkddfilesuffix です。<br><br>作業負荷が分散されている場合、パフォーマンスを向上させるために、プラグインは、バックアップホスト間でバックアップ対象を分散できます。このような場合は、バックアップホストそれぞれに対して作業負荷ファイルが作成されます。 |
| schedtype       | 入力: const char | バックアップスケジュールの種類: 完全、差分、累積増分。                                                                                                                                                                         |
| wlkddfilesuffix | 入力: const char | 作業負荷ファイル名が一意になるようにするために、プラグインが名前に付加する必要があるサフィックス。                                                                                                                                                    |
| snapid          | 入力: const char | 一意のスナップショット ID。プラグインは、カスタムのスナップショット ID を作成できます。                                                                                                                                                      |

出力: NetBackup PSF がこの API を呼び出した後、プラグインは作業負荷ファイルを、指定したディレクトリのバックアップホストごとに作成する必要があります。



## 戻り値

表 8-10 戻り値

| 値                            | 説明                                 |
|------------------------------|------------------------------------|
| AAPI_EOK                     | 成功 (Success)                       |
| AAPI_ESNAPSHOT_FAILED        | バックアップに失敗しました。スナップショットは作成されませんでした。 |
| AAPI_ECREATE_WORKLOAD_FAILED | バックアップに失敗しました。作業負荷ファイルは作成されませんでした。 |

すべての戻り値のリストは、p.90 の「[NetBackup Parallel Streaming Framework API の戻り値](#)」を参照してください。

## aapi\_flush

### 概要

```
int aapi_flush(aapi_pgn_object_handle_t object_handle)
```

### 説明

プラグインは、オブジェクトデータをフラッシュします。

### パラメータ (Parameters)

表 8-11 API のパラメータ

| パラメータ         | 型         | 説明         |
|---------------|-----------|------------|
| object_handle | 入力: const | オブジェクトハンドル |

## 戻り値

表 8-12 戻り値

| 値         | 説明              |
|-----------|-----------------|
| AAPI_EOK  | 成功 (Success)    |
| AAPI_ESYS | システムコールに失敗しました。 |

すべての戻り値のリストは、p.90 の「[NetBackup Parallel Streaming Framework API の戻り値](#)」を参照してください。

# aapi\_get\_object\_prop

## 概要

```
static int
aapi_get_object_prop(
    const char                *pathname,
    aapi_object_info_v1_t     *objinfo,
    const char                *servername,
    const aapi_session_def_t   *session,
    const aapi_pgn_server_handle_t server_handle)
```

## 説明

プラグインは、パス名に基づいてオブジェクトを特定し、オブジェクトのプロパティを返します。

## パラメータ (Parameters)

表 8-13 API のパラメータ

| パラメータ           | 型              | 説明                    |
|-----------------|----------------|-----------------------|
| pathname        | 入力: const char | オブジェクトのパス             |
| servername      | 入力: const char | アプリケーションサーバー名         |
| セッション (session) | 入力: const      | セッション定義へのポインタ。        |
| server_handle   | 入力: const      | サーバーハンドル              |
| objinfo         | 出力             | オブジェクトのプロパティ構造へのポインタ。 |

## 戻り値

表 8-14 戻り値

| 値        | 説明           |
|----------|--------------|
| AAPI_EOK | 成功 (Success) |

すべての戻り値のリストは、p.90 の「[NetBackup Parallel Streaming Framework API の戻り値](#)」を参照してください。

## 概要

```
int  
aapi_get_object_prop_byname(  
    const aapi_pgn_server_handle_t server_handle,  
    const char                      *pathname,  
    aapi_object_info_v1_t          *objinfo)
```

## 説明

NetBackup PSF は、この API の `aapi_object_info_v1_t` を割り当てて、プラグインにオブジェクトの場所を送信します。プラグインは、パス名に基づいてオブジェクトを特定し、オブジェクトのプロパティを返します。

## パラメータ (Parameters)

表 8-15 API のパラメータ

| パラメータ                      | 型                           | 説明                    |
|----------------------------|-----------------------------|-----------------------|
| <code>server_handle</code> | 入力: <code>const</code>      | サーバーハンドル              |
| <code>pathname</code>      | 入力: <code>const char</code> | オブジェクトのパス             |
| <code>objinfo</code>       | 出力                          | オブジェクトのプロパティ構造へのポインタ。 |

## 戻り値

表 8-16 戻り値

| 値                           | 説明                           |
|-----------------------------|------------------------------|
| <code>AAPI_EOK</code>       | 成功 (Success)                 |
| <code>AAPI_EINTERNAL</code> | NetBackup プロセスで内部エラーが発生しました。 |

すべての戻り値のリストは、p.90 の「[NetBackup Parallel Streaming Framework API の戻り値](#)」を参照してください。

# aapi\_get\_server\_prop

## 概要

```
aapi_get_server_prop(  
    aapi_pgn_server_handle_t  server_handle,  
    aapi_server_info_t        *serverInfo)
```

## 説明

この API は、プラグインでサポートされる NetBackup の機能を提供します。p.20 の「サ  
ポート対象の NetBackup 機能」を参照してください。

完全バックアップを行うためには、プラグインで AAPI\_SRVC\_FULL\_BACKUP\_SUPPORT 機  
能がサポートされる必要があります。スケジュール形式が「完全」で、プラグインが  
AAPI\_SRVC\_FULL\_BACKUP\_SUPPORT 機能をサポートしない場合、フレームワークは  
AAPI\_ESCHED\_NOT\_SUPPORTED エラーを返します。

作業負荷で累積的または差分の増分バックアップがサポートされている場合、プラグイン  
では、それぞれ AAPI\_SRVC\_CINC\_BACKUP\_SUPPORT または  
AAPI\_SRVC\_INCR\_BACKUP\_SUPPORT の機能をサポートできます。

ポリシーに提示されたスケジュール形式が累積的または差分の増分バックアップで、プラ  
グインがこれらの機能をサポートしない場合、フレームワークは  
AAPI\_ESCHED\_NOT\_SUPPORTED エラーを返します。

プラグインが必須の NetBackup 機能のセットをサポートする必要があります。

## パラメータ (Parameters)

表 8-17 API のパラメータ

| パラメータ         | 型       | 説明                            |
|---------------|---------|-------------------------------|
| server_handle | 入力: int | サーバーハンドル                      |
| serverInfo    | 出力      | プラグインがサポートする<br>NetBackup 機能。 |

## 戻り値

表 8-18 戻り値

| 値        | 説明           |
|----------|--------------|
| AAPI_EOK | 成功 (Success) |

すべての戻り値のリストは、p.90 の「[NetBackup Parallel Streaming Framework API の戻り値](#)」を参照してください。

## aapi\_get\_snappath\_from\_realpath

### 概要

```
int  
aapi_get_snappath_from_realpath(  
    const aapi_pgn_server_handle_t server_handle,  
    const char *snapname,  
    const char *realpath,  
    char *snappath,  
    size_t snappath_size)
```

### 説明

NetBackup PSF は、各オブジェクトのバックアップ中にこの API を呼び出します。

プラグインは、バックアップするオブジェクトのスナップショット名 (snapname) と仮想パス (realpath) を受け取ります。プラグインは、スナップショット名を使用して、オブジェクトの仮想パスをそのオブジェクトのスナップショットからの対応するパス (snappath) に変換します。

プラグインは、オブジェクトのスナップショットパスを返します。

### パラメータ (Parameters)

表 8-19 API のパラメータ

| パラメータ         | 型              | 説明                             |
|---------------|----------------|--------------------------------|
| server_handle | 入力: const      | サーバーハンドル                       |
| snapname      | 入力: const char | 作業負荷ファイルに示されたスナップショット名。        |
| realpath      | 入力: const char | 作業負荷ファイルに示されたオブジェクトの仮想パス。      |
| snappath_size | 入力: size_t     | snappath 変数のバッファサイズ。           |
| snappath      | 出力: char       | プラグインは、このパスからオブジェクトをバックアップします。 |

## 戻り値

表 8-20          戻り値

| 値        | 説明           |
|----------|--------------|
| AAPI_EOK | 成功 (Success) |

すべての戻り値のリストは、p.90 の「[NetBackup Parallel Streaming Framework API の戻り値](#)」を参照してください。

# aapi\_open\_object

## 概要

```
int  
aapi_open_object(  
    const aapi_pgn_server_handle_t  server_handle,  
    const char                      *pathname,  
    aapi_object_mode_t               mode,  
    uint64_t                        bufferSize,  
    aapi_pgn_object_handle_t         *pobject_handle)
```

## 説明

プラグインがバックアップ用のオブジェクトを特定して開きます。

必要に応じて、再利用のために任意のメタデータを aapi\_pgn\_object\_handle\_s に追加できます。

## パラメータ (Parameters)

表 8-21          API のパラメータ

| パラメータ         | 型            | 説明                          |
|---------------|--------------|-----------------------------|
| server_handle | 入力: const    | サーバーハンドル                    |
| pathname      | 入力: const    | バックアップするオブジェクトまたはエンティティのパス。 |
| mode          | 入力: int      | オブジェクトを開くモード (AAPI_O_READ)  |
| bufferSize    | 入力: uint64_t | 現在、この値は 0 です。               |

| パラメータ          | 型  | 説明                               |
|----------------|----|----------------------------------|
| pobject_handle | 出力 | 開かれたオブジェクトを参照するオブジェクトハンドルへのポインタ。 |

## 戻り値

表 8-22 戻り値

| 値                 | 説明                              |
|-------------------|---------------------------------|
| AAPI_EOK          | 成功 (Success)                    |
| AAPI_EISDIR       | 要求されたオブジェクトはディレクトリです。           |
| AAPI_EFNOTFOUND   | 操作の完了に必要なファイルが見つかりません。          |
| AAPI_ESNAPDIRPATH | ディレクトリまたはパスを作成できないか、アクセスできません。  |
| AAPI_EMALLOC      | ジョブの実行に使用できるメモリまたはリソースが不足しています。 |
| AAPI_ESYS         | システムコールに失敗しました。                 |
| AAPI_EINVAL       | 1 つ以上の入力パラメータまたは引数が無効です。        |

すべての戻り値のリストは、p.90 の「[NetBackup Parallel Streaming Framework API の戻り値](#)」を参照してください。

## aapi\_open\_server

### 概要

```
int  
aapi_open_server(  
    const aapi_session_def_t *session,  
    const aapi_server_name_t  aapi_server_name,  
    const aapi_cred_t          *credentials,  
    aapi_pgn_server_handle_t *server_handle)
```

### 説明

NetBackup PSF は、検出、バックアップ、リストア、または後処理の間にこの API を呼び出します。

API は、アプリケーションサーバー名およびクレデンシャルの検証に使用できます。クレデンシャルでは、`username:password` のクリアテキスト (`AAPI_CRED_CLEAR`) 形式が使用されます。

API は、`aapi_pgn_server_handle_s` の種類のサーバーハンドルを割り当てます。このサーバーハンドルには、すべての API 呼び出しで再利用するために、カスタムプラグインメタデータを定義できます。セッションハンドルには、開いているサーバーが開始するプロセスを示す `session_type` 変数を設定します。

たとえば、プロセスは検出 (`AAPI_SESSION_TYPE_DISCOVERY`)、バックアップ (`AAPI_SESSION_TYPE_BACKUP`)、リストア (`AAPI_SESSION_TYPE_RESTORE`)、または後処理 (`AAPI_SESSION_TYPE_POSTPROCESS`) になる可能性があります。

`session_type` の値は、プロセス固有の検証をカスタマイズするために使用できます。

## パラメータ (Parameters)

表 8-23 API のパラメータ

| パラメータ                          | 型                      | 説明                                                                                                        |
|--------------------------------|------------------------|-----------------------------------------------------------------------------------------------------------|
| セッション ( <code>session</code> ) | 入力: <code>const</code> | セッションへのハンドル                                                                                               |
| <code>aapi_server_name</code>  | 入力: <code>const</code> | <code>application_typeapplication_server_name</code> 形式のアプリケーションサーバー名                                     |
| クレデンシャル                        | 入力: <code>const</code> | アプリケーションサーバーのクレデンシャル<br>( <code>username:password</code> のクリアテキスト<br>( <code>AAPI_CRED_CLEAR</code> ) 形式) |
| <code>server_handle</code>     | 出力: <code>const</code> | サーバーハンドル                                                                                                  |

## 戻り値

表 8-24 戻り値

| 値                                        | 説明                     |
|------------------------------------------|------------------------|
| <code>AAPI_ECREDS_NOT_AVAILABLE</code>   | クレデンシャルが利用できません。       |
| <code>AAPI_ECREDS_INVALID</code>         | クレデンシャルが無効です。          |
| <code>AAPI_ESERVER</code>                | サーバー名が無効です。            |
| <code>AAPI_ESERVER_NOT_RESPONDING</code> | アプリケーションサーバーが応答していません。 |



すべての戻り値のリストは、p.90 の「NetBackup Parallel Streaming Framework API の戻り値」を参照してください。

# aapi\_pgn\_claim

## 概要

```
int
aapi_pgn_claim(const aapi_server_name_t serverName)
```

## 説明

NetBackup PSF は、プラグインでアプリケーションの種類がサポートされるかどうかを確認するために、この API を呼び出します。

プラグインでアプリケーションの種類がサポートされる場合、検出、バックアップ、リストア、または後処理の要求を取得できます。

## パラメータ (Parameters)

表 8-25 API のパラメータ

| パラメータ      | 型         | 説明                                                   |
|------------|-----------|------------------------------------------------------|
| serverName | 入力: const | application_typeapplication_server_name の形式のサーバー名です。 |

## 戻り値

表 8-26 戻り値

| 値           | 説明                                 |
|-------------|------------------------------------|
| AAPI_EOK    | プラグインは application_type をサポートします。  |
| AAPI_ECLAIM | プラグインは application_type をサポートしません。 |

すべての戻り値のリストは、p.90 の「NetBackup Parallel Streaming Framework API の戻り値」を参照してください。

# aapi\_pgn\_init

## 概要

```
int
aapi_pgn_init(
    uint64_t      sdkVersion,
    const char    *path,
    aapi_pgn_api_t *API)
```

## 説明

NetBackup 並列ストリームフレームワーク (PSF) は、プラグインを初期化するためにこの API を呼び出します。この API は、検出、バックアップ、リストア、または後処理のために呼び出されます。

PSF はプラグインに最新 SDK バージョンを送信し、プラグインは、サポートする SDK の最新バージョンのメソッドテーブルを返します。

プラグインは、ビルドバージョン、ベンダーのバージョン、プラグインの接頭辞、プラグインのラベルについての情報を提供する必要があります。

## パラメータ (Parameters)

表 8-27 API のパラメータ

| パラメータ          | 型              | 説明                                                                                        |
|----------------|----------------|-------------------------------------------------------------------------------------------|
| sdkVersion     | 入力: uint64_t   | フレームワークがサポートする最新の SDK バージョンを示します。<br><br>プラグインの SDK バージョンは、これと等しいか、このバージョンより古い必要があります。    |
| path           | 入力: const char | NetBackup がプラグインをロードする場所です。<br><br>例:<br><code>/export/psf/sdk/1.0.0/bin/psfplugin</code> |
| aapi_pgn_api_t | 出力: API        | サポート対象 SDK バージョンのメソッドテーブルです。<br><br>ビルドバージョン、ベンダーのバージョン、プラグインの接頭辞、プラグインのラベルを示します。         |

## 戻り値

表 8-28 戻り値

| 値                   | 説明                                           |
|---------------------|----------------------------------------------|
| AAPI_EOK            | 成功 (Success)                                 |
| AAPI_EPLUGINVERSION | プラグインが、PSF によって指定されている SDK バージョンをサポートしていません。 |

すべての戻り値のリストは、p.90 の「[NetBackup Parallel Streaming Framework API の戻り値](#)」を参照してください。

## aapi\_postprocess

### 概要

```
int aapi_postprocess(  
    const aapi_pgn_server_handle_t server_handle,  
    const char *selections,  
    const char *backuphosts,  
    const char *wkldoutdir,  
    const char *wkdfilesuffix,  
    const char *schedtype,  
    const char *snapid)
```

### 説明

NetBackup PSF は、バックアップ処理の後でこの API を呼び出して、後処理を開始するようにプラグインに通知します。後処理で、プラグインはスナップショットを削除し、その他のクリーンアップ操作を実行できます。

### パラメータ (Parameters)

表 8-29 API のパラメータ

| パラメータ         | 型              | 説明                                                   |
|---------------|----------------|------------------------------------------------------|
| server_handle | 入力: const      | サーバーハンドル                                             |
| selections    | 入力: const char | 構成済みポリシーのバックアップ対象です。<br><br>複数のバックアップ対象は、スペースで区切ります。 |

| パラメータ          | 型              | 説明                                                                                                                                                                                                  |
|----------------|----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| wkldoutdir     | 入力: const char | プラグインが作業負荷ファイルを作成する出力ディレクトリ。<br><br>作業負荷ファイル名の形式は、backuphost.wlkdfilesuffix です。<br><br>作業負荷が分散されている場合、パフォーマンスを向上させるために、プラグインは、バックアップホスト間でバックアップ対象を分散できます。このような場合は、バックアップホストそれぞれに対して作業負荷ファイルが作成されます。 |
| schedtype      | 入力: const char | バックアップスケジュールの種類: 完全、差分、累積増分。                                                                                                                                                                        |
| wlkdfilesuffix | 入力: const char | 作業負荷ファイル名が一意になるようにするために、プラグインが名前に付加する必要があるサフィックス。                                                                                                                                                   |
| snapid         | 入力: const char | 一意のスナップショット ID。プラグインは、カスタムのスナップショット ID を作成できます。                                                                                                                                                     |

戻り値

表 8-30            戻り値

| 値        | 説明           |
|----------|--------------|
| AAPI_EOK | 成功 (Success) |

すべての戻り値のリストは、[p.90 の「NetBackup Parallel Streaming Framework API の戻り値」](#)を参照してください。

aapi\_read\_object

概要

```
int
aapi_read_object(
```

```
aapi_pgn_object_handle_t object_handle,
void *buf,
uint64_t length,
uint64_t offset,
uint64_t *bytesRead)
```

説明

NetBackup PSF はこの API を呼び出して、オブジェクトハンドル、データ読み取りを開始するオフセット、読み込むデータの長さを送信します。

プラグインは、オブジェクトハンドルによって参照されるオブジェクトデータを、指定されたオフセットから読み込みます。指定した長さのデータを読み込んだら、プラグインは提供されたバッファにデータを格納します。

プラグインは、オブジェクトから読み込まれた実際のバイト数を bytesRead 変数に格納して値を返します。

パラメータ (Parameters)

表 8-31 API のパラメータ

| パラメータ         | 型            | 説明                |
|---------------|--------------|-------------------|
| object_handle | 入力: const    | オブジェクトハンドル        |
| length        | 入力: uint64_t | 読み込まれるデータ量。       |
| offset        | 入力: uint64_t | 読み取りオフセット         |
| buf           | 出力: void     | 読み取りデータを保持するバッファ。 |
| bytesRead     | 出力: uint64_t | 読み込まれるバイト数。       |

戻り値

表 8-32 戻り値

| 値           | 説明                       |
|-------------|--------------------------|
| AAPI_EOK    | 成功 (Success)             |
| AAPI_EIO    | 入力出力エラーです。               |
| AAPI_ESYS   | システムコールに失敗しました。          |
| AAPI_EINVAL | 1 つ以上の入力パラメータまたは引数が無効です。 |

すべての戻り値のリストは、p.90 の「[NetBackup Parallel Streaming Framework API の戻り値](#)」を参照してください。

# aapi\_set\_object\_utimes

## 概要

```
int aapi_set_object_utimes(  
    const aapi_pgn_server_handle_t  server_handle,  
    const char                      *pathname,  
    const aapi_object_utimes_vl_t   *utime)
```

## 説明

NetBackup PSF は、この API を呼び出して、オブジェクトのパス名、最終アクセス時刻、および変更時刻をプラグインに送信します。

プラグインはオブジェクトの最終アクセス時刻と変更時刻を更新します。

## パラメータ (Parameters)

表 8-33 API のパラメータ

| パラメータ         | 型              | 説明                                      |
|---------------|----------------|-----------------------------------------|
| server_handle | 入力: const      | セッションのサーバーハンドル。                         |
| pathname      | 入力: const char | オブジェクトまたはエンティティをリストアするため、リストア操作で指定したパス。 |
| utime         | 入力: const      | 最終アクセス時刻と修正時刻。                          |

## 戻り値

表 8-34 戻り値

| 値         | 説明           |
|-----------|--------------|
| AAPI_EOK  | 成功 (Success) |
| AAPI_ESYS | 失敗 (Failed)  |

すべての戻り値のリストは、p.90 の「[NetBackup Parallel Streaming Framework API の戻り値](#)」を参照してください。

# aapi\_terminate

## 概要

```
int
aapi_terminate(void)
```

## 説明

NetBackup PSF は、検出、バックアップ、リストア、または後処理の終了を示すためにこの API を呼び出します。

PSF は、プラグインをアンロードする前にこの API を呼び出します。必要に応じて、プラグインをアンロードする前に、プラグインまたは作業負荷に固有のクリーンアップ操作を追加できます。

## パラメータ (Parameters)

この API にはパラメータはありません。

## 戻り値

表 8-35                      戻り値

| 値        | 説明                                             |
|----------|------------------------------------------------|
| AAPI_EOK | 成功 (Success)<br><br>メモ: API では、その他の戻り値は無視されます。 |

すべての戻り値のリストは、p.90 の「[NetBackup Parallel Streaming Framework API の戻り値](#)」を参照してください。

# aapi\_write\_object

## 概要

```
int
aapi_write_object(
    aapi_pgn_object_handle_t object_handle,
    void *buf,
    uint64_t length,
    uint64_t offset,
    uint64_t *bytesWritten)
```

## 説明

NetBackup PSF はこの API を呼び出して、オブジェクトハンドル、データ書き込みを開始するオフセット、書き込むデータの長さを送信します。

プラグインは、指定された特定の長さのバッファ内のデータを読み込みます。プラグインは、オブジェクトハンドルによって参照されたオブジェクトに、指定されたオフセットからデータを書き込みます。

プラグインは、オブジェクトに書き込まれたデータ量を `bytesWritten` 変数に格納して、値を返します。

## パラメータ (Parameters)

表 8-36 API のパラメータ

| パラメータ                      | 型                         | 説明              |
|----------------------------|---------------------------|-----------------|
| <code>object_handle</code> | 入力: <code>const</code>    | オブジェクトハンドル      |
| <code>buf</code>           | 入力: <code>void</code>     | データを保持しているバッファ。 |
| <code>length</code>        | 入力: <code>uint64_t</code> | 書き込まれるデータ量。     |
| <code>offset</code>        | 入力: <code>uint64_t</code> | 書き込みオフセット       |
| <code>bytesWritten</code>  | 出力: <code>uint64_t</code> | 書き込まれたバイト数。     |

## 戻り値

表 8-37 戻り値

| 値                        | 説明                       |
|--------------------------|--------------------------|
| <code>AAPI_EOK</code>    | 成功 (Success)             |
| <code>AAPI_EIO</code>    | 入力出力エラーです。               |
| <code>AAPI_ESYS</code>   | システムコールに失敗しました。          |
| <code>AAPI_EINVAL</code> | 1 つ以上の入力パラメータまたは引数が無効です。 |

すべての戻り値のリストは、p.90 の「[NetBackup Parallel Streaming Framework API の戻り値](#)」を参照してください。



## aapi\_restore\_preprocess

### 概要

```
int aapi_restore_preprocess(  
    aapi_handle_t server_handle,  
    const char *intent,  
    const char *config)
```

### 説明

NetBackup PSF は、プラグインがリストア前操作を実行することを許可するために、この API を呼び出します。

### パラメータ (Parameters)

表 8-38 API のパラメータ

| パラメータ         | 型              | 説明           |
|---------------|----------------|--------------|
| server_handle | 入力: const      | サーバーハンドル     |
| intent        | 入力: const char | 現在使用されていません。 |
| 構成 (config)   | 入力: const char | 現在使用されていません。 |

### 戻り値

表 8-39 戻り値

| 値        | 説明           |
|----------|--------------|
| AAPI_EOK | 成功 (Success) |

すべての戻り値のリストは、p.90 の「[NetBackup Parallel Streaming Framework API の戻り値](#)」を参照してください。

## aapi\_restore\_postprocess

### 概要

```
int aapi_restore_postprocess(  
    aapi_handle_t server_handle,  
    const char *intent,  
    const char *config)
```

## 説明

NetBackup PSF は、プラグインがリストア後操作を実行することを許可するために、この API を呼び出します。

## パラメータ (Parameters)

表 8-40 API のパラメータ

| パラメータ         | 型              | 説明           |
|---------------|----------------|--------------|
| server_handle | 入力: const      | サーバーハンドル     |
| intent        | 入力: const char | 現在使用されていません。 |
| 構成 (config)   | 入力: const char | 現在使用されていません。 |

## 戻り値

表 8-41 戻り値

| 値        | 説明           |
|----------|--------------|
| AAPI_EOK | 成功 (Success) |

すべての戻り値のリストは、p.90 の「[NetBackup Parallel Streaming Framework API の戻り値](#)」を参照してください。

# NetBackup Parallel Streaming Framework API の戻り値

次の表に、NetBackup Parallel Streaming Framework API のすべての戻り値を一覧表示します。

表 8-42 NetBackup Parallel Streaming Framework API の戻り値

| 戻り値                          | 説明                                 |
|------------------------------|------------------------------------|
| AAPI_ECLAIM                  | プラグインは application_type をサポートしません。 |
| AAPI_ECREATE_WORKLOAD_FAILED | バックアップに失敗しました。作業負荷ファイルは作成されませんでした。 |
| AAPI_ECREDTS_INVALID         | クレデンシヤルが無効です。                      |
| AAPI_ECREDTS_NOT_AVAILABLE   | クレデンシヤルが利用できません。                   |

| 戻り値                         | 説明                                           |
|-----------------------------|----------------------------------------------|
| AAPI_EEXIST                 | 操作の完了に失敗しました。オブジェクトはすでに存在します。                |
| AAPI_EFNOTFOUND             | 操作の完了に必要なファイルが見つかりません。                       |
| AAPI_EINTERNAL              | <b>NetBackup</b> プロセスで内部エラーが発生しました。          |
| AAPI_EINVAL                 | 1 つ以上の入力パラメータまたは引数が無効です。                     |
| AAPI_EIO                    | 入力出力エラーです。                                   |
| AAPI_EISDIR                 | 要求されたオブジェクトはディレクトリです。                        |
| AAPI_EMALLOC                | ジョブの実行に使用できるメモリまたはリソースが不足しています。              |
| AAPI_EOK                    | 成功 (Success)                                 |
| AAPI_EPLUGINVERSION         | プラグインが、PSF によって指定されている SDK バージョンをサポートしていません。 |
| AAPI_ESERVER                | サーバー名とクレデンシャルは有効です。                          |
| AAPI_ESERVER_NOT_RESPONDING | アプリケーションサーバーが応答していません。                       |
| AAPI_ESNAPDIRPATH           | ディレクトリまたはパスを作成できないか、アクセスできません。               |
| AAPI_ESNAPSHOT_FAILED       | バックアップに失敗しました。スナップショットは作成されませんでした。           |
| AAPI_ESYS                   | システムコールに失敗しました。                              |

# ログとトラブルシューティング

この章では以下の項目について説明しています。

- [プラグインのログの有効化](#)

## プラグインのログの有効化

このセクションには、次のトピックが含まれます。

- [About NetBackup logs](#)
- [「NetBackup プラグインのログについて」](#)

### NetBackup ログについて

NetBackup は、NetBackup サーバーおよびクライアントのさまざまな処理についての情報を、ログディレクトリに記録します。NetBackup 管理者は、ログに含める詳細を決めるため、ログレベルを構成できます。ログレベルの範囲は 0 から 5 で、数が大きいほどログは詳細になります。

BigData ポリシー操作のすべてのログは、次の場所のサブディレクトリに書き込まれます。

- Windows の場合: `<install_path>\NetBackup\logs`
- UNIX の場合: `/usr/openv/netbackup/logs`

---

**メモ:** ログが書き込まれるようにするには、ログのフォルダがすでに存在する必要があります。フォルダが存在しない場合は、手動で作成する必要があります。

---

ログおよびログ記録について詳しくは、次を参照してください。

- [『Veritas NetBackup ログリファレンスガイド』](#)

## ■ 「グローバルログレベルについて」

## NetBackup プラグインのログについて

次の表に、NetBackup プラグインで利用可能なログに関する情報を一覧表示します。

表 9-1 NetBackup プラグインで利用可能なログ

| ログディレクトリ                                                                                                        | ログメッセージが関連する事項                           | ログが利用可能な場所 |
|-----------------------------------------------------------------------------------------------------------------|------------------------------------------|------------|
| Windows の場合:<br><install_path>/NetBackup/logs/lpVMutil<br>UNIX の場合:<br>/usr/openv/netbackup/logs/lpVMutil       | ポリシーの構成および検証                             | バックアップホスト  |
| Windows の場合:<br><install_path>/NetBackup/logs/nbaapidiscv<br>UNIX の場合:<br>/usr/openv/netbackup/logs/nbaapidiscv | PSF フレームワーク、検出、事前処理と<br>事後処理、およびプラグインのログ | バックアップホスト  |
| Windows の場合:<br><install_path>/NetBackup/logs/bpbirm<br>UNIX の場合:<br>/usr/openv/netbackup/logs/bpbirm           | ポリシー検証、バックアップ、およびリスト<br>ア操作              | メディアサーバー   |
| Windows の場合:<br><install_path>/NetBackup/logs/lpbkar<br>UNIX の場合:<br>/usr/openv/netbackup/logs/lpbkar           | バックアップ処理                                 | バックアップホスト  |
| Windows の場合:<br><install_path>/NetBackup/logs/tar<br>UNIX の場合:<br>/usr/openv/netbackup/logs/tar                 | リストアプロセス                                 | バックアップホスト  |

NetBackup PSF SDK API では、他の NetBackup プロセスと一貫性がある方法でプラグインの情報がログ記録されます。プラグインでは、`aapi_session_def_t->sd_log` 関数を使用して NetBackup と同じファイルにメッセージをログ記録します。たとえば、検出アプリケーションとプラグインの検出セッションのログメッセージは、照合されて、`<install_path>/NetBackup/logs/nbaapidiscv` ディレクトリに配置されている同じファイルに記録されます。統合ログメッセージには、関連する情報が順番に表示され、プ

ラグインの問題をトラブルシューティングするのに役立ちます。ログメッセージにはタイムスタンプとプロセス ID が含まれており、複数のプロセスが同時に実行されるときに便利です。

## ログファイルのデバッグレベル

ログファイルにはさまざまなデバッグレベルを適用でき、プラグイン開発者に適用可能なレベルは `aapi_common.h` ファイルで定義します。sd\_log メソッドで、次のレベルの重大度パラメータのいずれかを使用します。トラブルシューティングにログを活用できるようにするために、適切なデバッグ重大度レベルを使用するようにします。

次のリストに、ログレベルとその説明を示します。

| ログレベルの値                      | 説明           |
|------------------------------|--------------|
| <code>AAPI_ESINFO</code>     | 情報を示します。     |
| <code>AAPI_ESWARN</code>     | 警告を示します。     |
| <code>AAPI_ESDEBUG</code>    | デバッグを示します。   |
| <code>AAPI_ESERROR</code>    | エラーを示します。    |
| <code>AAPI_ESCRITICAL</code> | 重大なエラーを示します。 |

ログレベルを最小に設定した場合でも、重大なエラー、エラー、および警告は記録されます。

ログレベルについて詳しくは、次の記事を参照してください。

[「グローバルログレベルについて」](#)

## プラグインのログの有効化

検出、バックアップ、またはリストア操作用にすでに作成されている `NetBackup` の関連ログにエントリを追加するように、プラグインを設定できます。プラグインでは、セッションオブジェクトが提供する `sd_log` メソッドを呼び出す必要があります。これにより、操作に固有のログ方式が識別されます。セッションオブジェクトは、初期の `aapi_pgn_open_server API` で利用可能で、プラグインが作業負荷と通信するために使用します。

次の例は、ログを有効にするために、プラグインでセッションオブジェクトを使用する方法を示します。

```
const aapi_session_def_t *g_session_definition;  
//holds the session object globally  
  
void setGlobalSessionDef(const aapi_session_def_t * sd)  
{  
    g_session_definition = sd;  
}
```

```
}

const aapi_session_def_t * getGlobalSessionDef() {
return g_session_definition;
}

#define INFO getGlobalSessionDef(), AAPI_ESINFO, __FUNCTION__
#define WARNING getGlobalSessionDef(), AAPI_ESWARN, __FUNCTION__
#define DEBUG getGlobalSessionDef(), AAPI_ESDEBUG, __FUNCTION__
#define ERROR getGlobalSessionDef(), AAPI_ESERROR, __FUNCTION__
#define CRITICAL getGlobalSessionDef(), AAPI_ESCRITICAL, __FUNCTION__
#define SAMPLEPGN_LOG(x) LogMsg x;

//open_server implementation gets the session object stored
in the global session instance

pgn_open_server

(const aapi_session_def_t *session, const aapi_server_name_t
aapi_server_name,
const aapi_cred_t *credentials, aapi_pgn_server_handle_t *sh)

{
    setGlobalSessionDef(session);
    SAMPLEPGN_LOG(DEBUG, "entering");
    ...
}

//The wrapper log method is used to write to NetBackup logs.
//This method can either directly use the global session object, or
//accept the session instance as an input parameter for log writing

void LogMsg
(const aapi_session_def_v1_t *sd, aapi_severity_v1_t severity_level,
char *function, char *fmt, ...)

{
    //format the log message
    //log to NetBackup logs using session's sd_log method
    sd->sd_log("PLUGIN_APPTYPE_VERSION", formatted_message,
severity_level);
}
```

ログコードの実装について詳しくは、「[サンプルプラグインについて](#)」のトピックを参照してください。

プラグインの `sd_log` の呼び出しの最初のパラメータとして、一意かつ特定の文字列 (理想的にはプラグインの種類とバージョン) を使用することをお勧めします。ここで指定した文字列は、各ログメッセージの先頭に追加されます。これにより、ログファイルでプラグインに固有のエントリを識別しやすくなります。



## 記号

<workload> クレデンシヤル

追加 33

バックアップ 12

ホワイトリスト

バックアップホスト 29

リストア 13

作成

BigData バックアップポリシー 29

削除

バックアップホスト 27

用語 14

追加

バックアップホスト 27