



Red Hat Data Grid 8.5

Data Grid のアップグレード

Data Grid を 8.5 にアップグレードする

Red Hat Data Grid 8.5 Data Grid のアップグレード

Data Grid を 8.5 にアップグレードする

法律上の通知

Copyright © 2024 Red Hat, Inc.

The text of and illustrations in this document are licensed by Red Hat under a Creative Commons Attribution–Share Alike 3.0 Unported license ("CC-BY-SA"). An explanation of CC-BY-SA is available at

<http://creativecommons.org/licenses/by-sa/3.0/>

. In accordance with CC-BY-SA, if you distribute this document or an adaptation of it, you must provide the URL for the original version.

Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, the Red Hat logo, JBoss, OpenShift, Fedora, the Infinity logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

Linux[®] is the registered trademark of Linus Torvalds in the United States and other countries.

Java[®] is a registered trademark of Oracle and/or its affiliates.

XFS[®] is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL[®] is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js[®] is an official trademark of Joyent. Red Hat is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The OpenStack[®] Word Mark and OpenStack logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.

概要

Data Grid クラスターをある 8.x バージョンから別のバージョンにアップグレードします。ローリングアップグレードを実行して、Data Grid が互換性のためにデータを変換するダウンタイムまたはオフラインアップグレードを回避できます。

目次

RED HAT DATA GRID	3
DATA GRID のドキュメント	4
DATA GRID のダウンロード	5
多様性を受け入れるオープンソースの強化	6
第1章 DATA GRID 8 のアップグレードに関する注意点	7
1.1. DATA GRID 8.5 へのアップグレード	7
第2章 DATA GRID SERVER クラスターのローリングアップグレードの実行	11
2.1. ターゲット DATA GRID クラスターの設定	11
2.2. ターゲットクラスターへのデータの同期	12
第3章 キャッシュストア間のデータの移行	14
3.1. キャッシュストアマイグレーター	14
3.2. キャッシュストアマイグレーターの設定	14
3.3. DATA GRID キャッシュストアの移行	20

RED HAT DATA GRID

Data Grid は、高性能の分散型インメモリーデータストアです。

スキーマレスデータ構造

さまざまなオブジェクトをキーと値のペアとして格納する柔軟性があります。

グリッドベースのデータストレージ

クラスター間でデータを分散および複製するように設計されています。

エラスティックスケールリング

サービスを中断することなく、ノードの数を動的に調整して要件を満たします。

データの相互運用性

さまざまなエンドポイントからグリッド内のデータを保存、取得、およびクエリーします。

DATA GRID のドキュメント

Data Grid のドキュメントは、Red Hat カスタマーポータルで入手できます。

- [Data Grid 8.5 ドキュメント](#)
- [Data Grid 8.5 コンポーネントの詳細](#)
- [Data Grid 8.5 でサポートされる構成](#)
- [Data Grid 8 機能のサポート](#)
- [Data Grid で非推奨の機能](#)

DATA GRID のダウンロード

Red Hat カスタマーポータルで [Data Grid Software Downloads](#) にアクセスします。



注記

Data Grid ソフトウェアにアクセスしてダウンロードするには、Red Hat アカウントが必要です。

多様性を受け入れるオープンソースの強化

Red Hat では、コード、ドキュメント、Web プロパティにおける配慮に欠ける用語の置き換えに取り組んでいます。まずは、マスター (master)、スレーブ (slave)、ブラックリスト (blacklist)、ホワイトリスト (whitelist) の 4 つの用語の置き換えから始めます。この取り組みは膨大な作業を要するため、用語の置き換えは、今後の複数のリリースにわたって段階的に実施されます。詳細は、[Red Hat CTO である Chris Wright のメッセージ](#) をご覧ください。

第1章 DATA GRID 8 のアップグレードに関する注意点

Data Grid 8 のバージョンから別のバージョンにアップグレードする前に、このセクションの詳細を確認してください。

1.1. DATA GRID 8.5 へのアップグレード

以前のバージョンの Data Grid 8 から 8.5 へのアップグレードを正常に行うには、次の情報をお読みください。

削除

- Kryo および Protostuff マーシャラーは削除されました。
- 拡張統計情報モジュールは削除されました。
- JCache CDI はサポートされなくなりました。
- Spring 5.x および Spring Boot 2.x はサポートされなくなりました。
- Red Hat JBoss EAP モジュールは提供されなくなりました。

クラスのパッケージの変更

次のクラスは別のパッケージに移動されました。

- `org.infinispan.util.TimeoutException` は `org.infinispan.commons.TimeoutException` に移動されました
- `org.infinispan.util.concurrent.IsolationLevel` は `org.infinispan.configuration.IsolationLevel` に移動されました

クラス、メソッド、プロパティの削除

次のメソッドとプロパティは削除されました。

- すべて: スキーマ < 10.0 の設定ファイルの解析
- 共通: `Start.priority` と `Stop.priority`
- コア: `AdvancedCache.with(ClassLoader)`, `AdvancedCache.getEvictionManager()`, `AdvancedCache.getAsyncInterceptorChain()`, `AdvancedCache.getComponentRegistry()`
- コア: `AnyScopeComponentFactory.construct(Class)`
- コア: `Cache.getListeners()`
- コア: `ConfigurationBuilder.classLoader()`, `ConfigurationBuilder.customInterceptors()`
- コア: `DataRehashedEvent.getUnionConsistentHash()`
- コア: `EmbeddedCacheManager.getListeners()`, `EmbeddedCacheManager.getTransport()`, `EmbeddedCacheManager.removeCache(String)`, `EmbeddedCacheManager.getGlobalComponentRegistry()`
- コア: `FlagAffectedCommand.setFlags`, `FlagAffectedCommand.addFlags`, `FlagAffectedCommand.hasFlag`
- コア: `IntCacheStream.filterKeySegments(Set<Integer>)`

- コア: **JMXStatisticsConfiguration**
- コア: **GlobalConfiguration.DEFAULT_MARSHALL_VERSION**
- コア: **LocalizedCacheTopology.getDistributionForSegment(int)**
- コア: **NamedComponentFactory**
- コア: **PersistenceUtil.loadAndStoreInDataContainer**
- コア: **TopologyChangedEvent.getConsistentHashAtStart()** and **TopologyChangedEvent.getConsistentHashAtEnd()**
- コア: **Transport.invokeRemotely()** **Transport.waitForView()** **Transport.backupRemotely()** **AbstractTransport**
- コア: **ValueMatcher.nonExistentEntryCanMatch**
- コア: **WriteCommand.updateStatusFromRemoteResponse**
- コア: 状態遷移プール
- カウンター: **SyncStrongCounter** と **SyncWeakCounter**
- Hot Rod クライアント: **org.infinispan.client.hotrod.marshall.ProtoStreamMarshaller**
- サーバーとクライアント: キーストア証明書のパスワード
- RocksDB ストア: 有効期限キューのサイズ
- リモートストア: トランスポートファクトリーと `maxIdle`

メトリクス

names-as-tags が **true** に設定されている場合、JGroups およびクロスサイトメトリクスの名前が変更され、クラスターの名前とサイトの名前はメトリクス名に表示されなくなり、タグとして表示されます。

name-as-tags が **false** に設定されている場合、メトリクスがチャンネルに基づいて命名され、同じ目的のメトリクスが複数作成されます。

```
# TYPE vendor_jgroups_xsite_frag4_get_number_of_sent_fragments gauge
# HELP vendor_jgroups_xsite_frag4_get_number_of_sent_fragments Number of sent fragments
vendor_jgroups_xsite_frag4_get_number_of_sent_fragments{cluster="xsite",node="..."} 0.0
# TYPE vendor_jgroups_cluster_frag4_get_number_of_sent_fragments gauge
# HELP vendor_jgroups_cluster_frag4_get_number_of_sent_fragments Number of sent fragments
vendor_jgroups_cluster_frag4_get_number_of_sent_fragments{cluster="cluster",node="..."} 2.0
```

name-as-tags を **true** に設定すると、メトリクスが簡素化され、クラスター名とサイト名がタグとして表示されます。

```
# TYPE vendor_jgroups_frag4_get_number_of_sent_fragments gauge
# HELP vendor_jgroups_frag4_get_number_of_sent_fragments Number of sent fragments
vendor_jgroups_frag4_get_number_of_sent_fragments{cache_manager="default",cluster="xsite",node="..."} 0.0
vendor_jgroups_frag4_get_number_of_sent_fragments{cache_manager="default",cluster="cluster",node="..."} 2.0
```

SecurityManager

Java Development Kit (JDK) 17 での SecurityManager の削除による非推奨に合わせて、SecurityManager との統合はサポートされなくなりました。

Jakarta と Java EE

- Data Grid は **jakarta.*** パッケージのみを使用するように更新されました。
- レガシー **javax.*** パッケージが必要な場合は、Data Grid 8.4.x を使用する必要があります。
- **jakarta.*** と **javax.*** の両方のパッケージが含まれていた移行用の ***-jakarta** jar は削除されました。

Hot Rod クライアントのデフォルト

Data Grid では、Hot Rod クライアントのプロパティに変更が導入されました。

ssl_hostname_validation

デフォルト値が **true** の新しいプロパティ **ssl_hostname_validation** が追加されました。このプロパティにより、[RFC 2818](#) に基づいた TLS ホスト名の検証が有効になります。

さらに、ホスト名の検証が有効になっている場合は、**sni_host_name** の設定が必須になりました。

表1.1 デフォルトのプロパティの変更

プロパティ	Data Grid 8.5	以前のバージョン
connect_timeout	2000 ミリ秒/2 秒	60000 ミリ秒/60 秒
socket_timeout	2000 ミリ秒/2 秒	60000 ミリ秒/60 秒
max_retries	3	10
min_evictable_idle_time	180000 ミリ秒/3 分	1800000 ミリ秒/30 分
sasl_mechanism	SCRAM-SHA-512	SCRAM-SHA-256

Search

Data Grid 8.5 では、検索に次の変更が導入されています。

インデックス作成

- **property** 属性が削除されました。
- **auto-config** 属性が削除されました。
- **index** 属性が削除されました。

アノテーションのインデックス作成

- Hibernate Search 5 アノテーションはサポートされなくなりました。
- **@ProtoDoc** アノテーションは非推奨になりました。

セキュリティ

PrincipalRoleMapper は、グループとユーザープリンシパルの両方ではなく、グループにのみ適用されるようになりました。古い動作を使用するには、**cache-manager.security.authorization.group-only-mapping=false** 設定を使用します。

PrincipalRoleMapper API の詳細は、[ロールおよび権限マッパー](#) を参照してください。

散在 (scattered) キャッシュの削除

散在キャッシュモードは削除されました。代わりに分散キャッシュを使用してください。キャッシュモードの詳細は、[キャッシュモード](#) を参照してください。

グローバル状態

デフォルトでは、永続的なグローバル状態で、シャットダウンが正常に行われなかったことを示すダングリングロックファイルが見つかった場合、Data Grid は起動しません。グローバル状態の **unclean-shutdown-action** 設定を行うことで、デフォルトの動作を変更できます。詳細は、[グローバル永続OperationException](#) を参照してください。

Persistence

availability-interval のデフォルト値が 30 秒に増加されました。以前のデフォルトは 1 秒でした。

Soft-index ファイルストア

セグメント数を計算するには、以前のようにインデックスセグメント設定を使用する代わりに、キャッシュセグメントの数のみが使用されます。

RESP エンドポイント

Data Grid 8.5 では、RESP エンドポイントに次の変更が導入されています。

- RESP エンドポイントキャッシュでは、キーストレンジメディアタイプが **application/octet-stream** であることが必要になりました。
- RESP エンドポイントで使用されるキャッシュ設定にデフォルトの有効期限を適用できます。

クライアントリスナー削除イベントの伝播

CacheEventConverter の新しい **includeOldValue** メソッドに関する変更をサポートするために、クライアントリスナー削除イベントは、イベントが値を削除しない場合でも、伝播されるようになりました。



注記

削除イベントにはデフォルトでは値は含まれません。

NearCache SPI の更新

以前の NearCache SPI では、同時更新によって古いエントリがニアキャッシュに保存される可能性があるという問題がありました。この問題に対処するために、NearCache SPI が更新されました。

最低でも 8.1 からのアップグレード

8.0 からアップグレードする場合は、最初に 8.1 にアップグレードする必要があります。Data Grid 8.0 の永続データは、新しいバージョンのバイナリではありません。この非互換性の問題に対処するために、Data Grid 8.2 以降では、クラスター起動時に Data Grid 8.1 から既存の永続キャッシュストアを自動的に変換します。ただし、Data Grid は、Data Grid 8.0 からキャッシュストアを変換しません。

第2章 DATA GRID SERVER クラスターのローリングアップグレードの実行

Data Grid クラスターのローリングアップグレードを実行して、ダウンタイムやデータの損失なしにバージョン間で変更し、Hot Rod プロトコルを介してデータを移行します。

2.1. ターゲット DATA GRID クラスターの設定

アップグレードする予定の Data Grid バージョンを使用するクラスターを作成してから、リモートキャッシュストアを使用してソースクラスターをターゲットクラスターに接続します。

前提条件

- ターゲットクラスターに必要なバージョンの Data Grid Server ノードをインストールします。



重要

ターゲットクラスターのネットワークプロパティはソースクラスターのネットワークプロパティが重複していないことを確認します。JGroups トランスポート設定でターゲットおよびソースクラスターの一意の名前を指定する必要があります。環境に応じて、異なるネットワークインターフェイスとポートオフセットを使用して、ターゲットクラスターとソースクラスターを分離することもできます。

手順

- ターゲットクラスターがソースクラスターに接続できるリモートキャッシュストア設定を JSON 形式で作成します。
ターゲットクラスターのリモートキャッシュストアは、Hot Rod プロトコルを使用して、ソースクラスターからデータを取得します。

```
{
  "remote-store": {
    "cache": "myCache",
    "shared": true,
    "raw-values": true,
    "security": {
      "authentication": {
        "digest": {
          "username": "username",
          "password": "changeme",
          "realm": "default"
        }
      }
    }
  },
  "remote-server": [
    {
      "host": "127.0.0.1",
      "port": 12222
    }
  ]
}
```

2. Data Grid コマンドラインインターフェイス (CLI) または REST API を使用して、リモート キャッシュストア設定をターゲットクラスターに追加し、ソースクラスターに接続できるようにします。

- CLI: ターゲットクラスターで **migrate cluster connect** コマンドを使用します。

```
[/containers/default]> migrate cluster connect -c myCache --file=remote-store.json
```

- REST API: **rolling-upgrade/source-connection** メソッドを使用して、ペイロードにリモートストア設定が含まれる POST リクエストを呼び出します。

```
POST /rest/v2/caches/myCache/rolling-upgrade/source-connection
```

3. 移行するキャッシュごとに直前の手順を繰り返します。
4. すべての要求の処理を開始するために、クライアントをターゲットクラスターに切り替えま
 - a. クライアント設定をターゲットクラスターの場所で更新します。
 - b. クライアントを再起動します。



重要

インデックスキャッシュを移行する必要がある場合は、まず内部の **__protobuf_metadata** キャッシュを移行して、ソースクラスターで定義された .proto スキーマがターゲットクラスターにも存在するようにする必要があります。

関連情報

- [リモートキャッシュストア設定スキーマ](#)

2.2. ターゲットクラスターへのデータの同期

ターゲット Data Grid クラスターを設定してソースクラスターに接続する場合、ターゲットクラスターはリモートキャッシュストアを使用してクライアント要求を処理し、オンデマンドでデータをロードできます。データをターゲットクラスターに完全に移行して、ソースクラスターの使用を停止できるようにするには、データを同期します。この操作はソースクラスターからデータを読み取り、ターゲットクラスターに書き込みます。データは、ターゲットクラスターのすべてのノードに並行して移行され、各ノードはデータのサブセットを受け取ります。ターゲットクラスターに移行する各キャッシュの同期を実行する必要があります。

前提条件

- 適切な Data Grid バージョンでターゲットクラスターを設定している。

手順

1. ターゲットクラスターに移行する各キャッシュと Data Grid コマンドラインインターフェイス (CLI) または REST API との同期を開始します。
 - CLI: **migrate cluster synchronize** コマンドを使用します。

```
migrate cluster synchronize -c myCache
```

- REST API: POST リクエストと共に **?action=sync-data** パラメーターを使用します。

```
POST /rest/v2/caches/myCache?action=sync-data
```

操作が完了すると、Data Grid はターゲットクラスターにコピーされたエントリーの合計数で応答します。

2. ターゲットクラスター内の各ノードをソースクラスターから切断します。

- CLI: **migrate cluster disconnect** コマンドを使用します。

```
migrate cluster disconnect -c myCache
```

- REST API: DELETE リクエストを呼び出します。

```
DELETE /rest/v2/caches/myCache/rolling-upgrade/source-connection
```

次のステップ

ソースクラスターからすべてのデータを同期すると、ローリングアップグレードプロセスが完了します。ソースクラスターの使用を停止できるようになりました。

第3章 キャッシュストア間のデータの移行

Data Grid は、キャッシュストア間で永続化されたデータを移行するための Java ユーティリティを提供します。

Data Grid をアップグレードする場合、メジャーバージョン間の機能相違点は、キャッシュストア間の後方互換性を許可しません。**StoreMigrator** を使用してデータを変換し、ターゲットバージョンとの互換性を持つことができます。

たとえば、Data Grid 8.0 にアップグレードすると、デフォルトのマージャーが Protostream に変更になります。以前の Data Grid バージョンでは、キャッシュストアはバイナリー形式を使用し、マージングする変更との互換性がありません。つまり、Data Grid 8.0 は、以前の Data Grid バージョンでキャッシュストアから読み込むことができません。

他の場合は、Data Grid のバージョンが、JDBC Mixed および Binary ストアなどのキャッシュストア実装を非推奨または削除します。このような場合は、**StoreMigrator** を使用して異なるキャッシュストア実装に変換できます。

3.1. キャッシュストアマイグレーター

Data Grid は、最新の Data Grid キャッシュストア実装のデータを再作成する CLI **migrate store** コマンドを提供します。

ストアマイグレーターは、以前のバージョンの Data Grid のキャッシュストアをソースとして取得し、キャッシュストアの実装をターゲットとして使用します。

ストアマイグレーターを実行すると、**EmbeddedCacheManager** インターフェイスを使用して定義したキャッシュストアタイプでターゲットキャッシュが作成されます。ストアマイグレーターは、ソースストアからメモリーにエントリーをロードし、それらをターゲットキャッシュに配置します。

ストアマイグレーターを使用すると、あるタイプのキャッシュストアから別のストアにデータを移行することもできます。たとえば、JDBC 文字列ベースのキャッシュストアから、SIFS キャッシュストアに移行できます。



重要

ストアマイグレーターは、セグメント化されたキャッシュストアから以下へはデータを移行できません。

- 非セグメント化されたキャッシュストア。
- セグメント数が異なるセグメント化されたキャッシュストア。

3.2. キャッシュストアマイグレーターの設定

migrator.properties ファイルを使用して、ソースおよびターゲットのキャッシュストアのプロパティを設定します。

手順

1. **migrator.properties** ファイルを作成します。
2. **migrator.properties** ファイルを使用して、ソースおよびターゲットのキャッシュストアのプロパティを設定します。

→ [migrator.properties](#) 接続種を以て、マージャー、キャッシュストアの全設定プロパティ、に追加します。

- a. **source.** 接頭辞をソースキャッシュストアの全設定プロパティに追加します。

ソースキャッシュストアの例

```
source.type=SOFT_INDEX_FILE_STORE
source.cache_name=myCache
source.location=/path/to/source/sifs
source.version=<version>
```



重要

セグメント化されたキャッシュストアからデータを移行するには、**source.segment_count** プロパティを使用してセグメント数も設定する必要があります。セグメント数は、Data Grid 設定の **clustering.hash.numSegments** と一致させる必要があります。キャッシュストアのセグメント数が対応するキャッシュのセグメント数と一致しないと、Data Grid がキャッシュストアからデータを読み取ることができません。

- b. **target.** 接頭辞をターゲットキャッシュストアの全設定プロパティに追加します。

ターゲットキャッシュストアの例

```
target.type=SINGLE_FILE_STORE
target.cache_name=myCache
target.location=/path/to/target/sfs.dat
```

3.2.1. キャッシュストアマイグレーターの設定プロパティ

ソースおよびターゲットのキャッシュストアを **StoreMigrator** プロパティで設定します。

表3.1 キャッシュストアタイププロパティ

プロパティ	説明	必須/オプション
-------	----	----------

プロパティ	説明	必須/オプション
type	<p>ソースまたはターゲットのキャッシュストアのタイプを指定します。</p> <p>.type=JDBC_STRING</p> <p>.type=JDBC_BINARY</p> <p>.type=JDBC_MIXED</p> <p>.type=LEVELDB</p> <p>.type=ROCKSDB</p> <p>.type=SINGLE_FILE_STORE</p> <p>.type=SOFT_INDEX_FILE_STORE</p> <p>.type=JDBC_MIXED</p>	必須

表3.2 一般的なプロパティ

プロパティ	説明	値の例	必須/オプション
cache_name	バックアップするキャッシュの名前。	.cache_name=myCache	必須
segment_count	<p>セグメンテーションを使用できるターゲットキャッシュストアのセグメント数。</p> <p>セグメント数は、Data Grid 設定の clustering.hash.num Segments と一致させる必要があります。キャッシュストアのセグメント数が対応するキャッシュのセグメント数と一致しないと、Data Grid がキャッシュストアからデータを読み取ることができません。</p>	.segment_count=256	任意
marshaller.class	カスタムマーシャラークラスを指定します。	カスタムマーシャラーを使用する場合に必要です。	marshaller.allow-list.classes

プロパティ	説明	値の例	必須/オプション
デシリアライズが許可される完全修飾クラス名のコンマ区切りリストを指定します。	任意	marshaller.allow-list.regexps	どのクラスのデシリアライズを許可するかを決定する正規表現のコンマ区切りリストを指定します。
任意	marshaller.externalizers	[id]:<Externalizer class> 形式で読み込むカスタム AdvancedExternalizer 実装のコンマ区切りリストを指定します。	任意

表3.3 JDBC プロパティ

プロパティ	説明	必須/オプション
dialect	基礎となるデータベースのダイアレクトを指定します。	必須
version	ソースキャッシュストアのマージャーバージョンを指定します。 以下のいずれかの値を設定します。 * Data Grid 7.2.x の場合は 8 * Data Grid 7.3.x の場合は 9 * Data Grid 8.0.x の場合は 10 * Data Grid 8.1.x の場合は 11 * Data Grid 8.2.x の場合は 12 * Data Grid 8.3.x の場合は 13	ソースストアにのみ必要です。
connection_pool.connection_url	JDBC 接続 URL を指定します。	必須
connection_pool.driver_class	JDBC ドライバーのクラスを指定します。	必須
connection_pool.username	データベースユーザー名を指定します。	必須
connection_pool.password	データベースユーザー名のパスワードを指定します。	必須

プロパティ	説明	必須/オプション
db.disable_upsert	データベース upsert を無効にします。	任意
db.disable_indexing	テーブルインデックスが作成されるかどうかを指定します。	任意
table.string.table_name_prefix	テーブル名の追加接頭辞を指定します。	任意
table.string.<id data timestamp>.name	列名を指定します。	必須
table.string.<id data timestamp>.type	列タイプを指定します。	必須
key_to_string_mapper	TwoWayKey2StringMapper クラスを指定します。	任意

注記

Binary キャッシュストアから古い Data Grid バージョンの移行には、以下のプロパティで **table.string.*** を **table.binary.*** に変更します。

- **source.table.binary.table_name_prefix**
- **source.table.binary.<id\data\timestamp>.name**
- **source.table.binary.<id\data\timestamp>.type**

```
# Example configuration for migrating to a JDBC String-Based cache store
target.type=STRING
target.cache_name=myCache
target.dialect=POSTGRES
target.marshaller.class=org.infinispan.commons.marshall.JavaSerializationMarshaller
target.marshaller.allow-list.classes=org.example.Person,org.example.Animal
target.marshaller.allow-list.regexps="org.another.example.*"
target.marshaller.externalizers=25:Externalizer1,org.example.Externalizer2
target.connection_pool.connection_url=jdbc:postgresql:postgres
target.connection_pool.driver_class=org.postgresql.Driver
target.connection_pool.username=postgres
target.connection_pool.password=redhat
target.db.disable_upsert=false
target.db.disable_indexing=false
target.table.string.table_name_prefix=tablePrefix
target.table.string.id.name=id_column
target.table.string.data.name=datum_column
target.table.string.timestamp.name=timestamp_column
target.table.string.id.type=VARCHAR
target.table.string.data.type=bytea
```

```
target.table.string.timestamp.type=BIGINT
target.key_to_string_mapper=org.infinispan.persistence.keymappers.
DefaultTwoWayKey2StringMapper
```

表3.4 RocksDB プロパティ

プロパティ	説明	必須/オプション
location	データベースディレクトリを設定します。	必須
圧縮	使用する圧縮タイプを指定します。	任意

```
# Example configuration for migrating from a RocksDB cache store.
source.type=ROCKSDB
source.cache_name=myCache
source.location=/path/to/rocksdb/database
source.compression=SNAPPY
```

表3.5 SingleFileStore プロパティ

プロパティ	説明	必須/オプション
location	キャッシュストア .dat ファイルが含まれるディレクトリを設定します。	必須

```
# Example configuration for migrating to a Single File cache store.
target.type=SINGLE_FILE_STORE
target.cache_name=myCache
target.location=/path/to/sfs.dat
```

表3.6 SoftIndexFileStore プロパティ

プロパティ	説明	値
必須/オプション	location	データベースディレクトリを設定します。
必須	index_location	データベースインデックスディレクトリを設定します。

```
# Example configuration for migrating to a Soft-Index File cache store.
target.type=SOFT_INDEX_FILE_STORE
target.cache_name=myCache
target.location=path/to/sifs/database
target.index_location=path/to/sifs/index
```

3.3. DATA GRID キャッシュストアの移行

ストアマイグレーターを実行して、あるキャッシュストアから別のキャッシュストアにデータを移行します。

前提条件

- Data Grid CLI を入手している。
- ソースおよびターゲットのキャッシュストアを設定する **migrator.properties** ファイルを作成している。

手順

- **migrate store -p /path/to/migrator.properties** CLI コマンドを実行します。