



InterSystems IRIS での DataMove の使用法

Version 2024.1
2024-06-03

InterSystems®, HealthShare Care Community®, HealthShare Unified Care Record®, IntegratedML®, InterSystems Caché®, InterSystems Ensemble®, InterSystems HealthShare®, InterSystems IRIS®, および TrakCare は、InterSystems Corporation の登録商標です。HealthShare® CMS Solution Pack™ HealthShare® Health Connect Cloud™, InterSystems IRIS for Health™, InterSystems Supply Chain Orchestrator™, および InterSystems TotalView™ For Asset Management は、InterSystems Corporation の商標です。TrakCare は、オーストラリアおよび EU における登録商標です。

ここで使われている他の全てのブランドまたは製品名は、各社および各組織の商標または登録商標です。

このドキュメントは、インターシステムズ社(住所: One Memorial Drive, Cambridge, MA 02142)あるいはその子会社が所有する企業秘密および秘密情報を含んでおり、インターシステムズ社の製品を稼動および維持するためにのみ提供される。この発行物のいかなる部分も他の目的のために使用してはならない。また、インターシステムズ社の書面による事前の同意がない限り、本発行物を、いかなる形式、いかなる手段で、その全てまたは一部を、再発行、複製、開示、送付、検索可能なシステムへの保存、あるいは人またはコンピュータ言語への翻訳はしてはならない。

かかるプログラムと関連ドキュメントについて書かれているインターシステムズ社の標準ライセンス契約に記載されている範囲を除き、ここに記載された本ドキュメントとソフトウェアプログラムの複製、使用、廃棄は禁じられている。インターシステムズ社は、ソフトウェアライセンス契約に記載されている事項以外にかかるソフトウェアプログラムに関する説明と保証をするものではない。さらに、かかるソフトウェアに関する、あるいはかかるソフトウェアの使用から起こるいかなる損失、損害に対するインターシステムズ社の責任は、ソフトウェアライセンス契約にある事項に制限される。

前述は、そのコンピュータソフトウェアの使用およびそれによって起こるインターシステムズ社の責任の範囲、制限に関する一般的な概略である。完全な参照情報は、インターシステムズ社の標準ライセンス契約に記載され、そのコピーは要望によって入手することができる。

インターシステムズ社は、本ドキュメントにある誤りに対する責任を放棄する。また、インターシステムズ社は、独自の裁量にて事前通知なしに、本ドキュメントに記載された製品および実行に対する代替と変更を行う権利を有する。

インターシステムズ社の製品に関するサポートやご質問は、以下にお問い合わせください:

InterSystems Worldwide Response Center (WRC)
Tel: +1-617-621-0700
Tel: +44 (0) 844 854 2917
Email: support@InterSystems.com

目次

InterSystems IRIS での DataMove の使用法	1
1 DataMove の概要	1
2 DataMove の制限事項	1
3 DataMove を使用する準備	2
4 DataMove ワークフロー	3
5 ミラー環境または ECP 環境での DataMove	3
6 DataMove API	4
6.1 DataMove ネームスペース・マッピングの作成と編集	4
6.2 DataMove の生成	6
6.3 DataMove の開始	7
6.4 DataMove の監視	7
6.5 ^DATAMOVE ユーティリティ	8
6.6 マッピング変更の有効化と DataMove の完了	14
6.7 移動元のグローバルの削除と DataMove の完了	15
6.8 その他の API 呼び出し	15
7 DataMove の状態	18
8 DataMove の例	19
 テーブル一覧	
テーブル 1: DataMove の State のリスト	18

InterSystems IRIS での DataMove の使用法

このドキュメントでは、DataMove プロセスを使用して、InterSystems IRIS® データ・プラットフォームのネームスペースに関連付けられている既存のデータを新しい場所に移動する方法を説明します。

重要 DataMove は、実働環境で使用する前に、特定のネームスペースおよびデータベースを使用してテスト環境でテストしておくことを強くお勧めします。

注釈 このドキュメントに記載されている 1 行のコードは、ページ・サイズやフォント・サイズによっては折り返されて複数行にまたがっている場合があります。

1 DataMove の概要

DataMove プロセスでは、以下の操作を実行することで、InterSystems IRIS ネームスペースに関連付けられている既存のデータを別のデータベースに移動できます。

- ・ ネームスペースに新しいマッピングを作成する。
- ・ マッピングの変更を分析し、移動する必要があるグローバルおよびグローバル添え字を計算する。
- ・ データを新しい 1 つ以上のデータベースにコピーする。
- ・ マッピングの変更を有効にする。

例えば、DataMove を使用すると、グローバルの一部または全部をネームスペースの既定のグローバル・データベースから別のデータベースに移動できます。DataMove プロセスは、ネームスペースのグローバルをルーチンとは別のデータベースに移動する、複数のデータベースにまたがるデータを分割する、アプリケーションの進化に伴って更新された設計上の決定事項に基づいて新しい場所にデータを移動するなどの操作に使用できます。DataMove を使用すると、アプリケーションでデータをアクティブに使用しながら、同じデータを移動し、マッピングを変更できます。

マッピングの詳細は、“システム管理ガイド”の“[ネームスペースへのグローバル、ルーチン、およびパッケージ・マッピングの追加](#)”を参照してください。

2 DataMove の制限事項

DataMove プロセスには以下の制限事項が適用されます。

- ・ DataMove は、同一インスタンス上のデータベース間でデータを移動するよう設計されています。ミラーリングされたシステムおよび ECP クライアントが接続されたシステムには、操作上の特別な注意が必要です。
- ・ DataMove は、拡張グローバル参照を使用するアプリケーションでは使用できません。アプリケーションでマッピングと拡張参照が混在している場合、データ統合は保証されません。
- ・ システムでジャーナリングが有効になっている必要があります。
- ・ ジャーナル・エラー発生時にフリーズするスイッチが設定されている必要があります。
- ・ データを移動できるのは、ローカル・データベース間およびミラーリングされたデータベース間のみです。
- ・ ミラーリングされたデータベースを含む DataMove では、プライマリ・ノードでのみ DataMove を実行できます。

- ・ DataMove がプライマリ・ミラーで実行されているときに、フェイルオーバーまたは DR ノードがプライマリになった場合、DataMove は新しいプライマリ・ノードで自動的に実行を継続します。

拡張グローバル参照の詳細は、“ObjectScript の使用法”の“[ネームスペース](#)”を参照してください。

3 DataMove を使用する準備

DataMove は、進行状況と状態を追跡するために複数のグローバルを作成して使用します。これらのグローバルは、ユーザが作成する必要がある 2 つのデータベースに格納されます。これらのデータベースは、IRISSYS データベースと同じ属性で作成する必要があります。これには、ファイルとディレクトリの許可およびリソース名が含まれます。**MaxSize** パラメータは 0 (無制限) にする必要があり、データベースはジャーナリングする必要があります。これらのデータベースは数 GB に拡大する可能性があるため、十分なディスク容量が割り当てられていることを確認します。2 つのデータベースは以下のとおりです。

IRISDATAMOVE – ミラーリングされたまたはミラーリングされていないシステムで、あらゆるタイプの DataMove を実行するのに必要です。

IRISDATAMOVEMIRROR – ミラーリングされたシステムで DataMove を実行する場合にのみ必要です。このデータベースはミラーリングされたデータベースとして作成する必要があり、フェイルオーバー、非同期、DR、レポート・メンバを含め、ミラーのすべてのメンバに存在する必要があります。

これらのデータベースは、CPF ファイルで作成および定義する必要があります。以下に、ミラーリングされたシステムで CPF ファイルに含める内容の例を示します。

```
[Databases]
IRISDATAMOVE=C:\irisdatamove\
IRISDATAMOVEMIRROR=C:\irisdatamovemirror\
```

以下に、**IRISDATAMOVEMIRROR** データベースに指定すべき属性の例を示します。

```
%SYS>d ^DATABASE
1) Create a database
2) Edit a database
3) List databases
4) Delete a database
5) Mount a database
6) Dismount a database
7) Compact globals in a database
8) Show free space for a database
9) Show details for a database
10) Recreate a database
11) Manage database encryptions
12) Return unused space for a database
13) Compact free space in a database
14) Defragment a database
15) Show background database tasks

Option? 9
Database directories? IRISDATAMOVEMIRROR
Device:
Right margin: 80 =>

-----
Directory:          c:\irisdatamovemirror\
MirrorDBName:      IRISDATAMOVEMIRROR
MaxSize:           0
Size:              31
Status:            Mounted/RW
BlockSize:        8192
ClusterMountMode:  0
ClusterMounted:    0
ExpansionSize:     0
LastExpansionTime: 03/03/2023 15:25:25
Mounted:           1
NewGlobalCollation: IRIS standard
NewGlobalGrowthBlock: 50
NewGlobalIsKeep:   0
```

```

GlobalJournalState:      Yes
NewGlobalPointerBlock:   16
ReadOnly:                0
ResourceName:            %DB_IRISSYS
MountedReadOnly:         0
EncryptedDB:             0
EncryptionKeyID:         0
Configured DB Name:      IRISDATAMOVEMIRROR
Mount Required At Startup: No

```

4 DataMove ワークフロー

DataMove ワークフローは、以下のフェーズで構成されています。

1. ネームスペースのマッピングに対する変更が一時ストレージ領域に保存されます。
2. 一連のデータの移動がマッピングから生成されます。
3. データの移動が、指定されたグローバルおよびデータベースに対して検証され、移動先のデータベースに十分なディスク空き容量があることが確認されます。何らかの問題が検出された場合、ユーザはそれを修正して、ワークフローを再開できます。
4. DataMove が開始されると、複数のバックグラウンド・ジョブが既存のデータを新しい場所にコピーします。データがコピーされると、他のバックグラウンド・ジョブがジャーナル・ファイル进行处理して、コピーされたデータに変更を適用します。
5. すべてのデータがコピーされ、ジャーナル・ファイルが適用されたら、DataMove はコピー元およびコピー先のグローバルで DataCheck を実行し、両方のデータベースでデータが同じであることを確認します。
6. DataCheck が完了すると、ネームスペースで新しいマッピングが有効化されます。
7. ネームスペースへの変更が正常に有効化された後は、新しい場所にコピーされた古いソース・データを削除できます。

DataMove では、すべての処理のログ・ファイル **DataMoveName.log** が **/mgr** ディレクトリに保持されます。また、以前のすべての操作のレコードが **/mgr** ディレクトリの **DataMove.log** ファイルにあります。

5 ミラー環境または ECP 環境での DataMove

ECP 環境またはミラー環境で DataMove を使用する場合、新しいネームスペース・マッピングが ECP クライアントおよびミラーで更新されていることを確認してください。ミラー環境の場合に、新規作成したミラーリングされたデータベースにデータを移動するには、これらのミラーリングされたデータベースをバックアップ・ミラーと非同期ミラーにも作成する必要があります。

1. DataMove を作成し、その DataMove で新しいマッピングを有効化する準備が整うまで DataMove を実行します。
2. フェイルオーバー・ミラーを使用している場合は、有効化する前にバックアップを DR ミラーに降格させる必要があります。
3. ECP クライアントを使用している場合は、有効化する前に、そのクライアント上のアプリケーションを停止するか、ECP システムをダウン状態にする必要があります。
4. 非同期ミラーを使用している場合は、有効化する前に、そこにあるデータにアクセスするすべてのアプリケーションを停止するか、そこに直接接続しているすべての ECP クライアントを停止する必要があります。
5. この時点で、新しい DataMove マッピングを有効化できます。

6. 非同期ミラー上のマッピングを、適用したばかりの新しい DataMove マッピングに一致するよう更新します。これには、以前のバックアップ・フェイルオーバー・ミラーも含まれます。また、IRIS サーバに接続されている ECP クライアントを更新します。そのためには `iris merge コマンド` を使用できます。
7. 以前のバックアップ・ミラーを DR ミラーからフェイルオーバー・ミラーに昇格させます。
8. ECP クライアントを、適用したばかりの新しい DataMove マッピングで更新します。
9. 非同期ミラーと ECP クライアントでのアプリケーションの再起動を許可します。

6 DataMove API

このセクションでは、DataMove ワークフローの各フェーズで必要な API 呼び出しについて詳しく説明します。

以下のガイドラインをメモしておいてください。

- ・ 特定の要件に合わせて、必要なスクリプトやユーザ・インタフェースを指定する必要があります。
- ・ 必要なマクロを利用するには、コードに `%syDataMove` ファイルをインクルードする必要があります。
- ・ コードでは、次の API 呼び出しに進む前に、各メソッドから返されるステータスを確認する必要があります。
- ・ 移動先データベースが存在しない場合は、DataMove API を呼び出す前に作成する必要があります。
- ・ ワークフローは、`%SYS` ネームスペースで実行する必要があります。

重要

DataMove 処理では大量のデータを移動および削除する可能性があり、完了までにかかなりの時間がかかることがあります。そのため、DataMove ワークフローの次の時点で、システムのフル・バックアップを作成することをお勧めします。

1. `DataMove.API.StartCopy()` を呼び出す直前
2. `DataMove.API.Activate()` を呼び出す直前
3. `DataMove.API.DeleteSourceGlobals()` を呼び出す直前

フル・バックアップには、`/mgr` ディレクトリとすべてのジャーナル・データが含まれます。これらのバックアップは、スケジュールされた定期的なバックアップに加えて作成する必要があります。

DataMove では大量のジャーナル・データが生成される可能性があるため、ジャーナルのアーカイブやページ・スケジュールを短い期間に設定できます。DataMove では、DataMove 処理で不要になったジャーナル・ファイルのみをページできます。DataMove の使用中にジャーナル・ファイルをページする場合は、システムが提供するジャーナル・ページ方法のみを使用してください。

6.1 DataMove ネームスペース・マッピングの作成と編集

6.1.1 `DataMove.API.MapInitialize(Name as %String, Namespaces As %String) As %Status`

新しい一連のマッピング編集用の一時ストレージ領域を初期化します。

引数：

- ・ `Name` は、DataMove の名前です。
- ・ `Namespaces` は、DataMove を実行するネームスペースのコンマ区切りリストです。

このメソッドは、あらゆる編集を加える前に呼び出す必要があり、指定したネームスペースに対してのみ有効です。複数のネームスペースで、同じ DataMove の一部としてデータを移動できます。

例：

```
Set Namespace = "SALES"
Set Status = ##Class(DataMove.API).MapInitialize("DMName",Namespace)
If $$$ISOK(Status) Write !,$SYSTEM.Status.GetErrorText(Status)
```

例：

```
Set Namespaces = "SALES,PROSPECTS"
Set Status=##Class(DataMove.API).MapInitialize("DMName",Namespaces)
If $$$ISOK(Status) Write !,$SYSTEM.Status.GetErrorText(Status)
```

6.1.2 DataMove.API.MapGlobalsCreate(Name as %String, Namespace As %String, GblName As %String, ByRef Properties As %String) As %Status

一時ストレージ領域に、このネームスペースの新しいグローバル・マッピングを作成します。このメソッドは、この DataMove に含める予定のマッピングの数に応じて、1 回以上呼び出すことができます。

引数：

- ・ Name は、DataMove の名前です。
- ・ Namespace は、DataMove を実行するネームスペースです。
- ・ GblName は、特定のデータベースにマッピングするグローバルの名前です。
- ・ Properties は、このマッピングに必要なプロパティ (特にこのマッピングで使用するデータベースの名前) の配列です。

GblName 引数を “A” に設定すると、このマッピングをグローバル ^A 全体に適用することが指定されます。この引数を “A(5):A(10)” に設定すると、このマッピングを ^A(5) から ^A(10) まで (ただし ^A(10) は含まない) の添え字付きのグローバル範囲に適用することが指定されます。

Properties 引数を、Properties ("Database") が “USER2” に設定される配列 Properties に設定すると、グローバル (またはグローバル範囲) がデータベース USER2 にマッピングされることが指定されます。

既存のマッピングを変更または削除するには、それぞれ MapGlobalsModify() メソッド、MapGlobalsDelete() メソッドを使用できます。詳細は、“クラス・リファレンス”を参照してください。

例：

```
Set Properties("Database")="DSTDB"

;Move ^X(100) up to but not including ^X(200) to database DSTDB
Set Status = ##Class(DataMove.API).MapGlobalsCreate("DMName","SALES","X(100):(200)",Properties)

;Move the entire ^INFO global to database DSTDB
Set Status = ##Class(DataMove.API).MapGlobalsCreate("DMName","SALES","INFO",Properties)

;Move all Data from the first subscript in ^BILLING
;(including the node ^BILLING itself) up to but not including ^BILLING(100)
;to database DSTDB
Set Status = ##Class(DataMove.API).MapGlobalsCreate("DMName","SALES","BILLING(BEGIN):(100)",Properties)

;Move the entire global ^PROSPECT
;from its currently mapped database to DSTDB
Set Status = ##Class(DataMove.API).MapGlobalsModify("DMName","SALES","PROSPECT",Properties)

;Move the entire global ^ORDERS from its currently mapped database
;back to the default database for the namespace SALES
Set Status=##Class(DataMove.API).MapGlobalsDelete("DMName","SALES","ORDERS")
```

グローバル範囲を定義する方法の詳細は、“システム管理ガイド”の“[グローバル・マッピング](#)”を参照してください。

6.2 DataMove の生成

6.2.1 DataMove.API.Generate(Name As %String, ByRef Properties As %String, ByRef Warnings As %String, ByRef Errors As %String) As Status

一時ストレージ領域のマッピング編集に基づいて新しい DataMove を作成します。

引数：

- ・ Name は、作成する DataMove の名前です。
- ・ Properties は、DataMove の作成に使用するオプションのプロパティの配列です。
- ・ Warnings は、DataMove の実行を阻止することはない競合を含んで返される配列です。
- ・ Errors は、DataMove の実行を阻止する競合を含んで返される配列です。

Properties が、Properties 引数として渡される配列の場合、次のようになります。

- ・ Properties("MaxMBPerMin") では、DataMove 処理で移動先データベースに移動できる 1 分あたりの最大 MB 数をオプションで指定します。これを 0 に設定すると、DataMove は可能な限り高速で実行されます。これを渡さない場合は、システムの既定値が使用されます。
- ・ Properties("MaxMBCheckPerMin") では、オプションで、DataMove の DataCheck 処理でチェックできる 1 分あたりの最大 MB 数を指定します。
これを 0 に設定すると、DataMove の DataCheck は可能な限り高速で実行されます。これを渡さない場合は、システムの既定値が使用されます。
- ・ Properties("Description") では、オプションで実行対象の DataMove の説明を指定します。
- ・ Properties("Flags") では、オプションで DataMove 処理を説明する以下のようなフラグを指定します。
 - － \$\$\$BitNoSrcJournal、ジャーナリングされないデータベースをコピーできます。
 - － \$\$\$BitCheckActivate、マッピングの変更を有効にする前に、ユーザ指定のルーチン \$\$\$CheckActivate^ZDATAMOVE() を呼び出して、アプリケーションのステータスを確認します。
- ・ Properties("LogFile") では、オプションで、既定値以外の場合にログ・ファイル名を指定します。

Warnings 配列には、移動するデータが別のネームスペースからもマッピングされている場合にマッピングのリストが含まれます。この配列には、グローバルまたはグローバル範囲の名前が添え字として付きます。

Errors 配列には、競合によってデータの移動が阻止されるマッピングのリストが含まれます。この配列には、グローバルまたはグローバル範囲の名前が添え字として付きます。

例：

```
Set Properties("Flags")= $$$BitCheckActivate
Set Status=##Class(DataMove.API).Generate("TEST",Properties,Warnings,Errors)
```

データベースのジャーナリングの詳細は、“データ整合性ガイド”の“[ジャーナリング](#)”を参照してください。

バッチ・モードでのプロセスの実行の詳細は、“専用のシステム/ツールおよびユーティリティ”の“[プロセス管理](#)”を参照してください。

6.3 DataMove の開始

6.3.1 DataMove.API.StartCopy(Name As %String) As %Status

DataMove コピーを開始します。これにより、実際のデータ移動が処理されます。

引数：

- ・ Name は、DataMove の名前です。

このメソッドによって DataMove の最初のコピーが一連のバックグラウンド・ジョブとして開始され、これらのジョブによって個々の範囲が移動元データベースから移動先データベースにコピーされます。それと同時に、コピー時に変更されるデータの移動先データベースにジャーナル・トランザクションが適用されます。

コピーが実際に始まる前に、StartCopy() によって Validate() および ValidateSizes() が呼び出され、DataMove が検証されます。検証でエラーが返された場合、StartCopy() は検証エラーを返し、DataMove を開始しません。ユーザはエラーを修正してから、StartCopy() を再度呼び出すことができます。

コピーが完了したら、StartCopy() は DataCheck を呼び出して、コピー元とコピー先のデータが同じであることを確認します。DataCheck が正常に完了したら、Activate() を呼び出すことができます。

6.4 DataMove の監視

6.4.1 DataMove.API.GetProperties(Name As %String, ByRef Properties as %String) As %Status

DataMove の現在のプロパティを返します。

引数：

- ・ Name は、DataMove の名前です。
- ・ Properties は、以下のような DataMove プロパティの配列です。
 - Properties("ExpandedState") - 状態の展開された外部形式。
 - Properties("JRNMBToApply") - 処理対象のジャーナル・ファイルの現在の MB 数 = 0。
 - Properties("MaxMBPerMin") - DataMove で移動が許可されている 1 分あたりの最大 MB 数。
 - Properties("MaxMBCheckPerMin") - DataMove の DataCheck で処理できる 1 分あたりの最大 MB 数。
 - Properties("MBCopied") - コピー済みの MB 数。DataMove の終了時、すべてのデータがコピーされ、Activate() が呼び出される前、コピーされた MB 数は、ソース・データベースのデータが DataMove の実行中に更新されるシステムの **MBToCopy** の MB 数とは一致しない可能性があります。
 - Properties("MBChecked") - DataCheck された MB 数。DataCheck が完了すると、この値はコピーされたデータのサイズより大きくなります。
 - Properties("MBToCopy") - コピー対象のおおよその MB 数。
 - Properties("State") - 移動の現在の状態。以下の表を参照してください。
 - Properties("StateExternal") - **State** プロパティの外部形式。
 - Properties("Status") - 発生したエラーの %Status の値。
 - Properties("Stop") - DataMove の停止状態：
 - ・ \$\$\$DMStopNone - 停止のシグナルなし。
 - ・ \$\$\$DMStopNormal - Activate() メソッドによって呼び出されたときに、停止のシグナルが送信されました。

- ・ `$$$DMStopShutdown` - 通常のシステム・シャットダウンにより停止のシグナルが送信されました。システム起動時に再起動するパラメータが設定されている場合、DataMove はシステム再起動時に再開します。
- ・ `$$$DMStopUser` - ユーザのメソッド `StopCopy()` の呼び出しにより、停止のシグナルが送信されました。コピーは、`StartCopy()` によって再開できます。
- ・ `$$$DMStopForce` - `Force()` によって停止のシグナルが送信されました。コピーは、`StartCopy()` によって再開できます。
- ・ `$$$DMStopErrorRecoverable` - エラーにより停止のシグナルが送信されました。エラーの修正後に、`StartCopy()` によってコピーを再開できます。
- ・ `$$$DMStopErrorUnrecoverable` - 回復不能エラーにより停止のシグナルが送信されました。これはジャーナル・エラーなど複数の異なる原因による可能性があります。ここで実行できる唯一の方法は、`Rollback()` を呼び出してから `StartCopy()` を呼び出して DataMove を再起動するか、`Delete()` を呼び出して DataMove を削除することです。

例：

```
%SYS>Set x=##Class(DataMove.API).GetProperties("ABC1",,Properties)
```

```
%SYS>Zwrite Properties
Properties("ExpandedState")="ActivateReady/Run"
Properties("JRNMBToApply")=0
Properties("MBChecked")=25622
Properties("MBCopied")=24722
Properties("MBToCopy")=24722
Properties("MaxMBCheckPerMin")=0
Properties("MaxMBPerMin")=0
Properties("State")=10
Properties("StateExternal")="ActivateReady"
Properties("Status")=1
Properties("Stop")=0
```

6.5 ^DATAMOVE ユーティリティ

^DATAMOVE ユーティリティは、システムに定義されているすべての DataMove の現在の状態を表示します。ユーザはこれを使用して DataMove の既定のパラメータを設定できます。複数の DataMove を同時に実行できます。

6.5.1 DataMove の既定のパラメータの設定

DataMove の既定のシステム・パラメータを使用して、システムの再起動時に実行する処理を指定できます。また、1 分あたりに移動するデータの量を制限することもできます。既定で、実行中の DataMove は可能な限り高速にデータをコピーします。ただし、これによりシステム上の他のユーザ・プロセスや、ミラー・システムの遅延に影響が及ぶ可能性があります。

```
%SYS>d ^DATAMOVE
1) Monitor DataMove
2) Edit DataMove settings
3) Exit
Option? 2
```

```
Setting the maximum number of MB DataMove is allowed to copy per minute
to 0 means that DataMove will copy the data as fast as it can. Changing
this setting will apply the new value to future DataMoves, and optionally
any currently running DataMoves. You can also change this setting through
the 'Metrics' display screen for a running DataMove without changing the
system default.
Maximum number of MB to copy per minute? 0 => 5000
Do you want to apply this change to existing Data Moves? Yes => Yes
```

DataMove の DataCheck でチェックできる 1 分あたりの最大 MB 数を 0 に設定すると、DataMove の DataCheck は可能な限り高速でデータをチェックします。この設定を変更すると、今後の DataMove に新しい値が適用され、オプション

で現在実行中の DataMove にも新しい値が適用されます。この設定は、システムの既定を変更することなく、実行中の DataMove DataCheck の [メトリック] 表示画面でも変更できます。

```
Maximum number of MB to check per minute? 0 => 8000
Do you want to apply this change to existing Data Moves? Yes => Yes
Confirm changes? Yes => Yes
DataMove settings updated
```

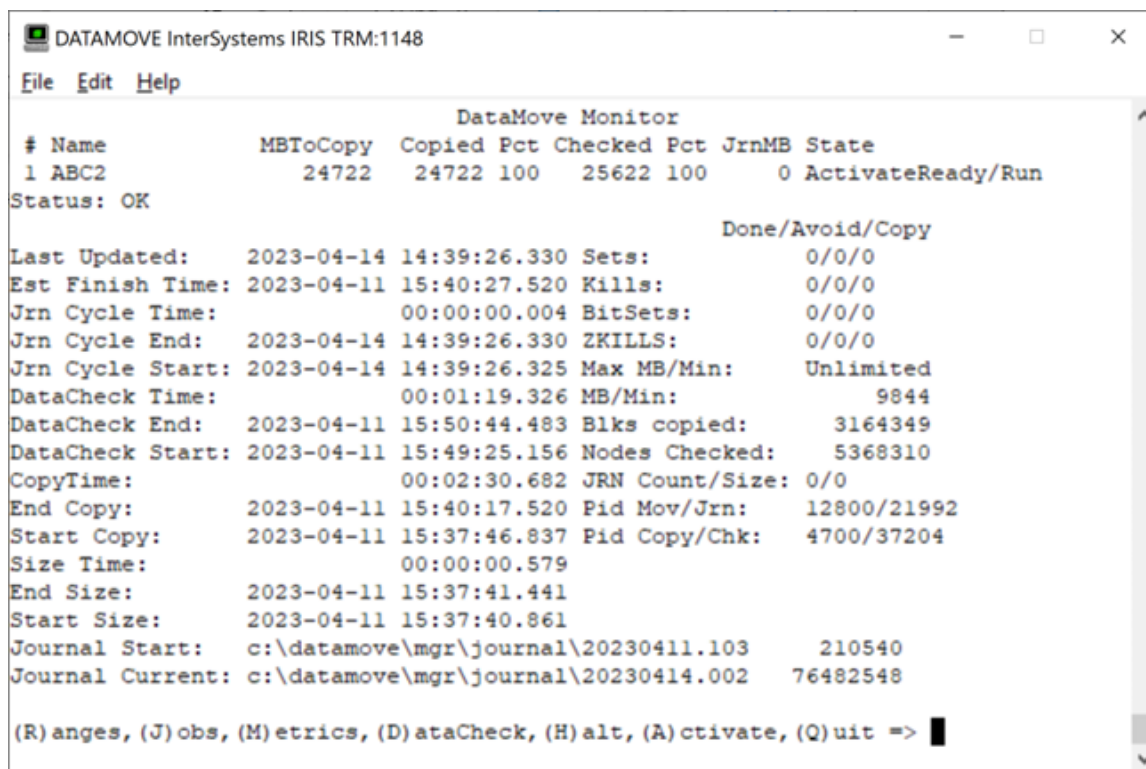
```
1) Monitor DataMove
2) Edit DataMove settings
3) Exit
```

6.5.2 DataMove の監視

Generate() メソッドを呼び出した後は、^DATAMOVE を使用して、移動の状態および移動対象として設定されている範囲を確認できます。

```
%SYS>d ^DATAMOVE
1) Monitor DataMove
2) Edit DataMove settings
3) Exit
Option? 1
```

これは、Generate() および StartCopy() を呼び出した後に表示される内容の例です。これは、DataMove 内のすべての移動対象範囲の累積統計データを示しています。



これらのフィールドは、以下の情報を示しています。

- ・ Jrn Cycle Time - ジャーナル・スキヤナまたはジャーナルの適用において、前回ジャーナルの末尾に達したときから、現在のジャーナル・ファイルの末尾に達するまでにかかった時間。
- ・ DataCheck Time - DataCheck の完了にかかった時間。
- ・ CopyTime - グローバルのコピーにかかった時間。
- ・ Size Time - グローバルのサイズの計算にかかった時間。
- ・ Done - ジャーナル・ファイル内で検出された各操作のうち、移動先データベースに適用されたものの総数。

- ・ **Avoid** - ジャーナル・ファイル内で検出された各操作のうち、スキップ可能だったものの総数。コピーがまだ開始していない場合、操作をスキップできます。コピーが開始していた場合は、移動先にまだコピーされていないグローバルの一部で操作が実行されます。
- ・ **Copy** - コピーの実行中に移動先データベースに適用された各操作の総数。
- ・ **Journal Start** - DataMove が開始されたジャーナル・ファイルとオフセット。
- ・ **Journal Current** - 現在のジャーナル・ファイルとオフセット。
- ・ **Max MB/Min** - DataMove でコピーが許可されている 1 分あたりの最大 MB 数。これを DataMove に設定するには、以下の 3 つの方法のいずれかを使用します。
 - DATAMOVE に指定されている DataMove の既定値
 - 以下に示している [メトリック] 画面
 - Modify() API
- ・ **MB/Min** - DataMove の生存期間中に DataMove がコピーした 1 分あたりの MB 数。
- ・ **Blks Copied** - コピーされたデータベース・ブロックの実際の数。
- ・ **Nodes Checked** - DataCheck されたグローバル・ノードの数。
- ・ **JRN Count/Size** - 対象の DataMove について、現在のジャーナル・ファイル、および StartCopy() が呼び出される前の前回のジャーナル・ファイルで検出されたトランザクションの数と合計サイズ。
- ・ **Pid Move/Jrn** - コピーおよびデジャーナルを処理する 2 つのプロセスのプロセス ID。
- ・ **Pid Copy/Chk** - マスタ DataMove プロセスとマスタ DataCheck プロセスのプロセス ID。

範囲オプションを選択した場合、移動対象のグローバルの範囲が 8 つあることがわかります。

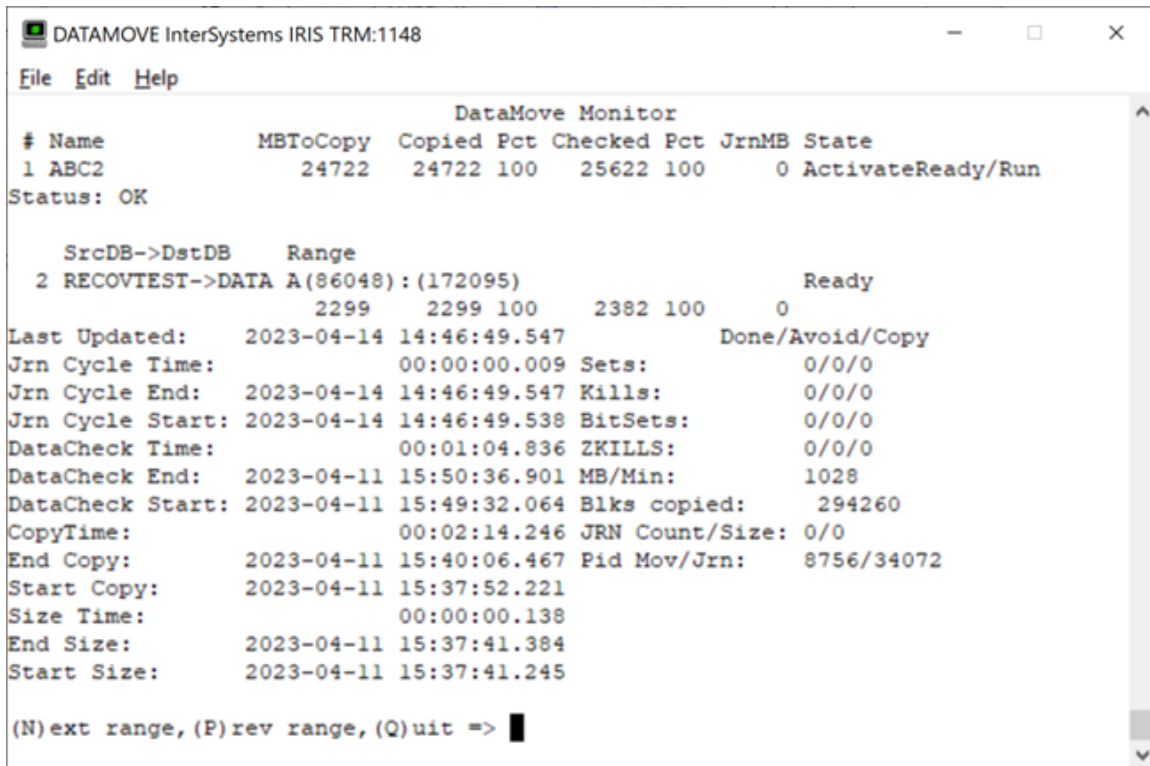
#	Name	MBToCopy	Copied	Pct	Checked	Pct	JrnMB	State
1	ABC2	24722	24722	100	25622	100	0	ActivateReady/Run

Status: OK

SrcDB->DstDB	Range	MBToCopy	Copied	Pct	Checked	Pct	JrnMB	State
1 RECOVTEST->DATA	A(BEGIN):(86048)	3308	3308	100	3450	100	0	Ready
2 RECOVTEST->DATA	A(86048):(172095)	2299	2299	100	2382	100	0	Ready
3 RECOVTEST->DATA	A(172095):(258142)	2014	2014	100	2091	100	0	Ready
4 RECOVTEST->DATA	A(258142):(344189)	1743	1743	100	1798	100	0	Ready
5 RECOVTEST->DATA	A(344189):(430236)	1633	1633	100	1671	100	0	Ready
6 RECOVTEST->DATA	A(430236):(516283)	1427	1427	100	1472	100	0	Ready
7 RECOVTEST->DATA	A(516283):(602330)	1348	1348	100	1402	100	0	Ready
8 RECOVTEST->DATA	A(602330):(688377)	1279	1279	100	1332	100	0	Ready

(N)ext, (P)rev, (E)xamine, (J)obs, (M)etrics, s(T)ats, (D)ataCheck, (H)alt, (Q)uit =>

範囲の詳細画面で、各範囲の詳細を確認できます。これらは上記で説明されていますが、選択した範囲にのみ適用されます。



```

DATAMOVE InterSystems IRIS TRM:1148
File Edit Help

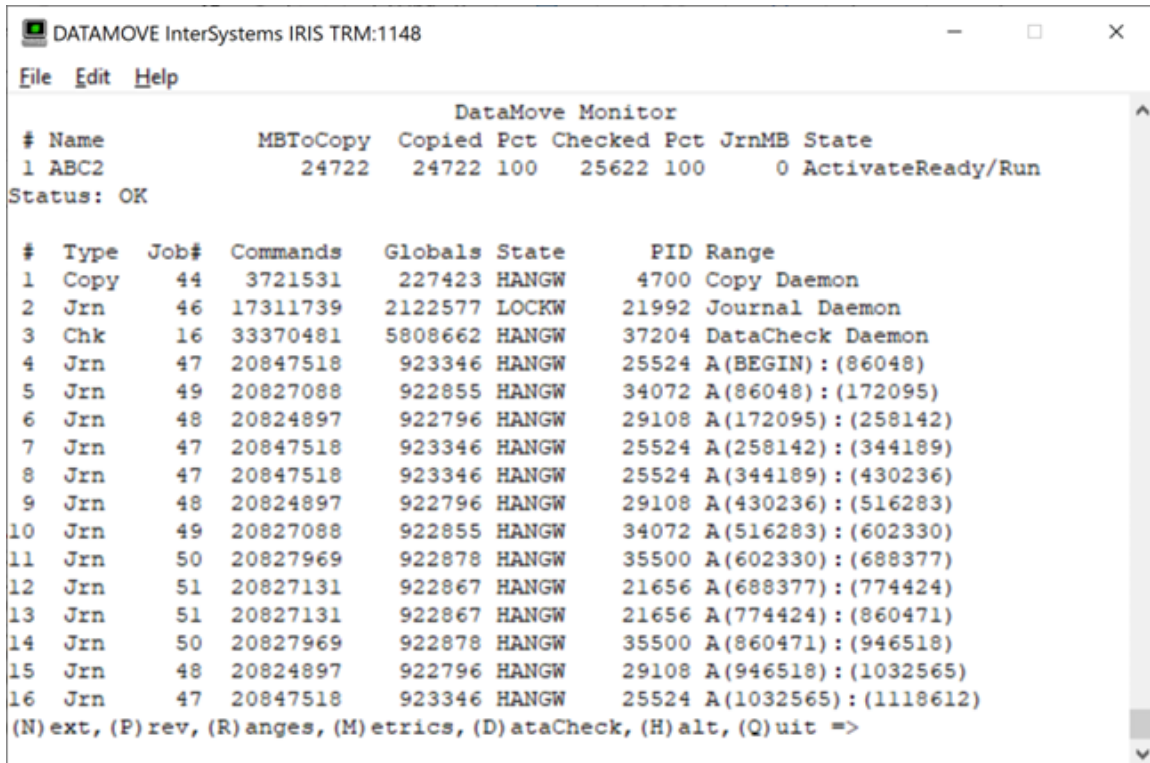
DataMove Monitor
# Name      MBToCopy  Copied Pct  Checked Pct  JrnMB  State
1 ABC2      24722    24722 100    25622 100    0 ActivateReady/Run
Status: OK

SrcDB->DstDB  Range
2 RECOVTEST->DATA A(86048):(172095)  Ready
      2299    2299 100    2382 100    0
Last Updated: 2023-04-14 14:46:49.547 Done/Avoid/Copy
Jrn Cycle Time: 00:00:00.009 Sets: 0/0/0
Jrn Cycle End: 2023-04-14 14:46:49.547 Kills: 0/0/0
Jrn Cycle Start: 2023-04-14 14:46:49.538 BitSets: 0/0/0
DataCheck Time: 00:01:04.836 ZKILLS: 0/0/0
DataCheck End: 2023-04-11 15:50:36.901 MB/Min: 1028
DataCheck Start: 2023-04-11 15:49:32.064 Blks copied: 294260
CopyTime: 00:02:14.246 JRN Count/Size: 0/0
End Copy: 2023-04-11 15:40:06.467 Pid Mov/Jrn: 8756/34072
Start Copy: 2023-04-11 15:37:52.221
Size Time: 00:00:00.138
End Size: 2023-04-11 15:37:41.384
Start Size: 2023-04-11 15:37:41.245

(N)ext range, (P)rev range, (Q)uit =>

```

ジョブ画面には、どのジョブがどのデータで実行されているかが示されます。



```

DATAMOVE InterSystems IRIS TRM:1148
File Edit Help

DataMove Monitor
# Name      MBToCopy  Copied Pct  Checked Pct  JrnMB  State
1 ABC2      24722    24722 100    25622 100    0 ActivateReady/Run
Status: OK

# Type  Job#  Commands  Globals  State  PID Range
1 Copy  44  3721531  227423  HANGW  4700 Copy Daemon
2 Jrn   46  17311739 2122577  LOCKW  21992 Journal Daemon
3 Chk   16  33370481 5808662  HANGW  37204 DataCheck Daemon
4 Jrn   47  20847518 923346  HANGW  25524 A(BEGIN):(86048)
5 Jrn   49  20827088 922855  HANGW  34072 A(86048):(172095)
6 Jrn   48  20824897 922796  HANGW  29108 A(172095):(258142)
7 Jrn   47  20847518 923346  HANGW  25524 A(258142):(344189)
8 Jrn   47  20847518 923346  HANGW  25524 A(344189):(430236)
9 Jrn   48  20824897 922796  HANGW  29108 A(430236):(516283)
10 Jrn  49  20827088 922855  HANGW  34072 A(516283):(602330)
11 Jrn  50  20827969 922878  HANGW  35500 A(602330):(688377)
12 Jrn  51  20827131 922867  HANGW  21656 A(688377):(774424)
13 Jrn  51  20827131 922867  HANGW  21656 A(774424):(860471)
14 Jrn  50  20827969 922878  HANGW  35500 A(860471):(946518)
15 Jrn  48  20824897 922796  HANGW  29108 A(946518):(1032565)
16 Jrn  47  20847518 923346  HANGW  25524 A(1032565):(1118612)
(N)ext, (P)rev, (R)anges, (M)etrics, (D)ataCheck, (H)alt, (Q)uit =>

```

“Copy”タイプのジョブは、コピー元からコピー先データベースにデータをコピーするプロセスです。“Jrn”タイプのジョブは、宛先データベースにジャーナル・トランザクションを適用します。“Chk”タイプのジョブは、コピー元データベースとコ

ピー先データベースの間で DataCheck を実行します。ここに表示されるジョブ番号、コマンド、グローバル、状態、およびプロセス ID は、JOBEXAM または管理ポータルのプロセス表示に示される同じ情報に対応しています。

メトリック画面には、システム上のアクティビティと、実行中の DataMove にそのアクティビティがどの程度関係しているかが示されます。

```

MIR2 InterSystems IRIS TRM:18800
File Edit Help

DataMove Monitor
# Name          MBToCopy  Copied Pct  Checked Pct  JrnMB  State
1 ABC2          20034    9684 48      0 0    1074 Copy/Run
Status: OK

Total      DataMove  DM %
Global references/Sec: 19905    19724    99
Global update references/Sec: 19185    19192    100
Physical reads/Sec: 8830    8869    100
Journal Entries/Sec: 19175    19192    100
MB Copied/Min Allowed/Current Unlimited 0
MB Checked/Min Allowed/Current Unlimited 0

Mirror Member Name      Member Type      Status      Dejournaling
USP5540STEVEC/MIR1     Disaster Recovery Connected 30 seconds behind

(R)anges, (J)obs, (D)ataCheck, (C)hange rate, (H)alt, (Q)uit =>

```

この例では、DataMove でコピーが許可される 1 分あたりの最大 MB 数が 0 (無制限) に設定されていることがわかります。また、DataMove がグローバル参照とジャーナル・エントリの大部分を占有していて、ミラー処理の遅延が増加する原因になっていることもわかります。

以下のように実行することで、DataMove の処理速度が低減されるよう制限できます。


```

MIR2 InterSystems IRIS TRM:18800
File Edit Help

DataMove Monitor
# Name          MBToCopy  Copied Pct  Checked Pct  JrnMB  State
1 ABC2          20034    17665  88         0    0    0 Copy/Run
Status: OK

                                Total      DataMove  DM %
Global references/Sec:         4746         4402    93
Global update references/Sec:  4349         4236    97
Physical reads/Sec:           3446         3287    95
Journal Entries/Sec:          4339         4236    98
MB Copied/Min  Allowed/Current  Unlimited    2394
MB Checked/Min Allowed/Current  Unlimited      0

Mirror Member Name      Member Type      Status      Dejournaling
USP5540STEVEC/MIR1      Disaster Recovery Connected 12 seconds behind

Max allowed MB Copied/Min is currently set to: 0-Unlimited
New value 0 (0=Unlimited) => 1200

```

これで、処理速度が更新され、メトリックが大幅に低減されました。

```

MIR2 InterSystems IRIS TRM:18800
File Edit Help

DataMove Monitor
# Name          MBToCopy  Copied Pct  Checked Pct  JrnMB  State
1 ABC2          20034    19385  97         0    0    0 Copy/Run
Status: OK

                                Total      DataMove  DM %
Global references/Sec:         192         1      1
Global update references/Sec:   15         0      0
Physical reads/Sec:            0         0      0
Journal Entries/Sec:           7         0      0
MB Copied/Min  Allowed/Current  1200      1153
MB Checked/Min Allowed/Current  Unlimited    0

Mirror Member Name      Member Type      Status      Dejournaling
USP5540STEVEC/MIR1      Disaster Recovery Connected Caught up

(R)anges, (J)obs, (D)ataCheck, (C)hange rate, (H)alt, (Q)uit =>

```

6.6 マッピング変更の有効化と DataMove の完了

6.6.1 DataMove.API.Activate(Name As %String, Display as %Boolean, Timeout As %Integer = 120, Force as %Integer=0) As %Status

DataMove を完了し、ネームスペースのマッピング変更を有効化します。Activate() は、DataCheck() が完了するまで実行されません。

引数：

- ・ Name は、DataMove の名前です。
- ・ Display は、進捗状況メッセージを確認する場合は 1 に設定する必要があります。
- ・ Timeout は、DataMove が実行を完了し、ジャーナル処理を適用する間、続行を待機する秒数です。
- ・ Force は、ジャーナルがキャッチアップされていない場合でもネームスペース・マッピングの有効化を開始します。Force フラグを設定すると、スイッチ 10 が設定されている間、追加で時間を費やして、有効化プロセスで最後のジャーナル・ファイルが適用されます。

このメソッドにより、DataMove バックグラウンド・ジョブが停止し、ジャーナル・ファイルの処理が完了し、マッピング変更が CPF に書き込まれ、マッピング変更が有効になります。また、他のプロセスが実行を妨げないように、スイッチ 10 が一時的に設定されます。

\$\$\$BitCheckActivate が設定されている場合、続行する前にこのメソッドによりユーザ指定のルーチン \$\$CheckActivate^ZDATAMOVE() が呼び出され、実行されます (存在する場合)。\$\$CheckActivate^ZDATAMOVE() から 1 が返されない場合、メソッドは新しいマッピングを有効にせずに終了し、DataMove は実行中のままになります。

注釈 スイッチ 10 が設定される直前に、ユーザ指定のルーチンが呼び出されます。このルーチンは、新しいマッピングを有効にする前に、ユーザが希望することをすべて実行できます。

スイッチ 10 の詳細は、“専用のシステム/ツールおよびユーティリティ” の “[スイッチの使用法](#)” を参照してください。

注釈 Activate() により DataMove の State プロパティがチェックされ、最初のコピーが完了し、DataCheck() が完了し、続行する前にジャーナル適用がキャッチアップされたことが確認されます。

6.6.2 ミラー・メンバと ECP クライアントでのマッピングの更新

システムで新しいネームスペース・マッピングを有効化したら、他のすべてのミラー・メンバと ECP クライアントでマッピングを更新できます。DataMove では、MGR ディレクトリに **DataMove_DataMoveName_Actions.cpf** という名前のファイルが生成されます。このファイルには、ミラーおよび ECP CPF ファイルにマージできる一連のアクションが含まれます。以下に、このファイルの形式の例を示します。

```
[Actions]
CreateMapGlobal:Namespace=SALES,Name=A(86048):(172095),Database=DATA
CreateMapGlobal:Namespace=SALES,Name=A(172095):(258142),Database=DATA
CreateMapGlobal:Namespace=SALES,Name=A(258142):(344189),Database=DATA
CreateMapGlobal:Namespace=SALES,Name=A(344189):(430236),Database=DATA
```

このファイルを詳しく見てみましょう。これには、システムの CPF ファイルに新たに適用された一連のマッピングが含まれている必要があります。このファイルを他のミラー・メンバと ECP クライアントにコピーし、**iris merge コマンド**を使用してマッピングをそれらのシステム上の CPF ファイルに挿入して有効化することができます。

6.7 移動元のグローバルの削除と DataMove の完了

コピーが完了し、更新されたネームスペースが有効化されると、アプリケーション・データが正常にコピーされ、マッピングが有効化されていることを確認する必要があります。これを確認した後は、移動元にある移動済みのグローバルの削除処理に進み、DataMove を完了することができます。

6.7.1 DataMove.API.DeleteSourceGlobals(Name As %String) As %Status

DataMove でコピーされたグローバルを移動元ディレクトリから削除します。

引数：

- ・ Name は、DataMove の名前です。

このメソッドにより、移動先データベースにコピー済みの、移動元データベース内のすべてのグローバルが削除されます。移動元のグローバルを削除するために、複数のプロセスが作成される場合があります。

6.7.2 DataMove.API.Finish(Name As %String) As %Status

DataMove プロセスを完了します。

引数：

- ・ Name は、DataMove の名前です。

このメソッドにより、ログ・ファイルに成功または失敗のメッセージが書き込まれ、ログ・ファイルが閉じられ、DataMove の State プロパティが `$$$DMStateDone` に設定されます。また、ログ・ファイルがファイル `DataMove.log` にコピーされます。このファイルは、システムで実行されたすべての DataMove の記録です。

6.7.3 DataMove.API.Delete(Name As %String) As %Status

DataMove プロセスをクリーンアップします。

引数：

- ・ Name は、DataMove の名前です。

このメソッドにより、DataMove が削除され、一時ストレージがクリーンアップされます。

削除したい DataMove 処理が開始されてから停止した場合は、まず `Rollback()` メソッドを呼び出して、移動されていたデータがあればそれをロールバックします。

6.8 その他の API 呼び出し

6.8.1 DataMove.API.Validate(Name As %String) As %Status

DataMove を検証します。

引数：

- ・ Name は、DataMove の名前です。

DataMove の検証には、指定したマッピングすべての確認、移動元および移動先のデータベースの確認、移動先のグローバルがまだ存在していないことの確認が含まれます。何らかのエラーが発生した場合は Status で報告されます。

`Validate()` は、`StartCopy()` の一部として呼び出されます。このメソッドを使用して、`StartCopy()` でコピーを実際に開始する前に DataMove を検証できます。

6.8.2 DataMove.API.Modify(Name as %String, byref Properties as %String) as %Status

既存のまたは実行中の DataMove に変更を加えます。

引数：

- ・ Name は、DataMove の名前です。
- ・ Properties は、変更するプロパティの配列で、以下が含まれる可能性があります。
 - － Properties("MaxMBPerMin") - DataMove 処理で移動先データベースに移動できる 1 分あたりの最大 MB 数。これを 0 に設定すると、DataMove は可能な限り高速で実行されます。実行中の DataMove に対してこれ呼び出して、処理速度を変更することができます。
 - － Properties("MaxMBCheckPerMin") - DataMove の DataCheck 処理でチェックできる 1 分あたりの最大 MB 数。これを 0 に設定すると、DataMove の DataCheck は可能な限り高速で実行されます。実行中の DataMove に対してこれ呼び出して、処理速度を変更することができます。

6.8.3 DataMove.API.ValidateSizes(Name As %String) As %Status

DataMove によってコピー対象に指定されたデータに対し、十分な容量が存在することを確認します。

引数：

- ・ Name は、DataMove の名前です。

DataMove のサイズの検証には、コピーするデータ量の特定、および移動先データベースに十分な容量があることの確認が含まれます。何らかのエラーが発生した場合は Status で報告されます。

ValidateSizes() は、StartCopy() の一部として呼び出されます。このメソッドを使用して、StartCopy() でコピーを実際に開始する前に DataMove のサイズ要件を検証できます。

6.8.4 DataMove.API.StopCopy(Name As %String) As %Status

DataMove コピー・ジョブを停止します。

引数：

- ・ Name は、DataMove の名前です。

このメソッドにより、DataMove コピー・バックグラウンド・ジョブが停止し、進行中のコピーを適切に停止できるようになります。DataMove の再起動は、StartCopy() を使用して行えます。

6.8.5 DataMove.API.Rollback(Name As %String) As %Status

DataMove をロールバックします。

引数：

- ・ Name は、DataMove の名前です。

このメソッドにより、DataMove コピー・ジョブによって移動先データベースにコピーされたグローバルがすべて削除されます。このメソッドは、DataMove の中止またはコピー・プロセスでのエラーからの回復に使用できます。

このメソッドを呼び出す前に StopCopy() を実行する必要があります。

ロールバックの実行後は、StartCopy() を使用してやり直すことも、Delete() を使用して Data Move を削除することもできます。

6.8.6 DataMove.API.RollbackMappings(Name As %String)

DataMove マッピングをロールバックします。

引数：

- ・ Name は、DataMove の名前です。

これにより、システムのマッピングが DataMove Name の実行前の状態に復元されます。DataMove Name の State が `$$$DMStateNSPActivateDone` である必要があります。この使用例として、DataMove でマッピングが有効化された後にアプリケーションのテストを開始し、何らかの問題が検出された場合が考えられます。マッピングが以前の値に復元されると、State は `$$$DMStateReady` に設定されます。その後で、`Rollback()` を呼び出すか (推奨)、`StartCopy()` を呼び出して有効化を再試行することができます。

6.8.7 DataMove.API.RollbackCopy(Name As %String, Warnings as %String, Errors as %String)

DataMove のコピーをロールバックします。

引数：

- ・ Name は、DataMove の名前です。
- ・ Warnings は、DataMove の実行を阻止することはない競合を含んで返される配列です。
- ・ Errors は、DataMove の実行を阻止する競合を含んで返される配列です。

これにより、Name-ROLLBACK という名前の DataMove が作成されます。この DataMove を実行すると、DataMove Name でコピーされたデータが、元の移動元ディレクトリに戻されます。DataMove Name の State は `$$$DMStateDeleteSrcGlobalsDone` である必要があります (移動元のグローバルはすべて `DeleteSourceGlobals()` メソッドによって削除されています)。このメソッドが完了すると、`StartCopy()` および `Activate()` を実行して、データを元のディレクトリに戻すことができます。これにより、DataMove Name の State は `$$$DMStateRollbackCopy` に設定されます。

6.8.8 Query DataMove.API.ListDMs(Names As %String, Flags as %Integer=0) As %Status

すべての DataMove およびそのプロパティのリストを返すクエリ。

引数：

- ・ Names は、DataMove のコンマ区切りのリストです。
- ・ フラグ -
 - 0 = DataMove レコードに記録されている現在の情報を返します。
 - 1 = すべての範囲をまとめた情報を返します。

6.8.9 Query DataMove.API.ListRanges(Name As %String, SrcDBs As %String = "*", DstDBs As %String = "*", Ranges As %String = "*", Flags = 0) As %Status

すべての DataMove 範囲およびそのプロパティのリストを返すクエリ。

引数：

- ・ Names は、DataMove の名前です。
- ・ SrcDBs - 移動元データベースのコンマ区切りのリスト
- ・ DstDBs - 移動先データベースのコンマ区切りのリスト
- ・ Ranges - 範囲のコンマ区切りのリスト

- ・ フラグ –
 - 0 = フォーマットされた時刻を返します。
 - 1 = \$h UTC 時刻として時刻を返します。

6.8.10 Query DataMove.API.ListProcesses(Name As %String) As %Status

すべての DataMove プロセスのリストを返すクエリ。

引数：

- ・ Name は、DataMove の名前です。

7 DataMove の状態

DataMove では、State プロパティを使用して、ワークフロー全体を通して進捗状況が記録されます。GetProperties() を使用することで、このプロパティを調べて、進捗状況を監視できます。問題が生じた場合は、クエリ ListDMs() を使用して、トラブルシューティングすることができます。

各 State の整数値は、%syDataMove.inc インクルード・ファイルで定義されています。

テーブル 1: DataMove の State のリスト

State	説明
\$\$\$DMStateCreate	DataMove は作成中です。
\$\$\$DMStateNotStarted	DataMove を作成するために Generate() が呼び出されました。
\$\$\$DMStateStarted	コピーを開始するために StartCopy() が呼び出されました。
\$\$\$DMStateSize	移動するデータのサイズが計算されています。
\$\$\$DMStateSizeDone	移動するデータのサイズが計算されました。
\$\$\$DMStateCopy	ソース・データが移動先データベースにコピーされています。
\$\$\$DMStateCopyDone	ソース・データの移動先データベースへのコピーが完了しました。
\$\$\$DMStateJournal	すべてのソース・データが移動先データベースにコピーされ、現在ジャーナルが移動先データベースに適用されています。DataCheck の実行が開始されるまで、この状態が維持されます。
\$\$\$DMStateDataCheck	DataMove プロセスは DataCheck を実行中です
\$\$\$DMStateReady	DataMove は DataCheck を完了し、有効化できる状態です。
\$\$\$DMStateNSPActivate	Activate() が呼び出されました。スイッチ 10 が設定され、新しいネームスペース・マッピングが有効化されています。

State	説明
\$\$\$DMStateNSPActivateDone	新しいネームスペース・マッピングが有効化されました。
\$\$\$DMStateDeleteSrcGlobals	DeleteSourceGlobals() が呼び出され、移動元のグローバルが削除されています。
\$\$\$DMStateDeleteSrcGlobalsDone	移動元のグローバルがすべて削除されました。
\$\$\$DMStateDone	Finish() が呼び出されました。すべてのデータが移動先データベースに移動され、ネームスペースが有効化され、最終的なログ・ファイルが更新されました。
\$\$\$DMStateRollback	Rollback() メソッドが呼び出され、コピーされたすべての移動先グローバルが削除されています。完了すると、State は \$\$\$DMStateNotStarted に設定されます。
\$\$\$DMStateRollbackCopy	RollbackCopy() メソッドが呼び出されました。この DataMove は完了し、ここからデータを前の場所に戻す新しい DataMove が作成されました。

8 DataMove の例

ObjectScript ルーチンは、DataMover という DataMove オブジェクトを使用して、ネームスペース LIVE のグローバル ABC および DEF をデータベース LIVE-CT に移動します。

この例は、デモンストレーションを目的としているため、包括的なエラー・チェックは含まれていません。各 API 呼び出しの後に返されるステータスを確認してください。

```
IRIS for Windows^MAC^^~Format=IRIS.S~^RAW
%RO on 22 Jun 2023    2:33 PM
DMEXAMPLE^MAC^^66647,52367.6489234^0
DMEXAMPLE
#include %occInclude
#include %syDataMove
/*
   This example creates a DataMove called "TEST" which moves pieces of the
   global ^A in namespace TESTDATA into 3 new databases.

   Assume the following entries in the CPF file, where the NEWDATA* databases
   are newly created empty databases, and the DATA database contains the
   entire global ^A.
.
  [Databases]
  DATA=c:\iris\mgr\data
  NEWDATA1=c:\iris\mgr\newdata1
  NEWDATA2=c:\iris\mgr\newdata2
  NEWDATA3=c:\iris\mgr\newdata3
.
  [Namespaces]
  TESTDATA=DATA
.
  The first subscript in ^A up to ^A(5000) will move into database NEWDATA1.
  ^A(5000) up to ^A(7000) will move into database NEWDATA2.
  ^A(7000) up to ^A(10000) is not moved and remains in the database DATA.
  ^A(10000) through the last subscript will move into database NEWDATA3.

  After the DataMove completes, the CPF file will have mappings added to it as follows:

  [MAP.TESTDATA]
  Global_A(BEGIN):(5000)=NEWDATA1
  Global_A(5000):(7000)=NEWDATA2
  Global_A(10000):(END)=NEWDATA3

*/
New
s DMName="TEST"
s Namespace="TESTDATA"
```

```

#;Initialize the namespace
s Status=##Class(DataMove.API).MapInitialize(DMName,Namespace)
i '$$$ISOK(Status) w !,$$SYSTEM.Status.GetErrorText(Status) q
#;Now create the 3 mappings for the DataMove
s Properties("Database")="NEWDATA1"
s Status=##Class(DataMove.API).MapGlobalsCreate(DMName,Namespace,"A(BEGIN):(5000)",.Properties)
i '$$$ISOK(Status) w !,$$SYSTEM.Status.GetErrorText(Status) q
s Properties("Database")="NEWDATA2"
s Status=##Class(DataMove.API).MapGlobalsCreate(DMName,Namespace,"A(5000):(7000)",.Properties)
i '$$$ISOK(Status) w !,$$SYSTEM.Status.GetErrorText(Status) q
s Properties("Database")="NEWDATA3"
s Status=##Class(DataMove.API).MapGlobalsCreate(DMName,Namespace,"A(10000):(END)",.Properties)
i '$$$ISOK(Status) w !,$$SYSTEM.Status.GetErrorText(Status) q
#;Generate the DataMove. Limit the maximum number of MB per minute the
#;DataMove operation is allowed to move to 1GB (60GB/hour)
k Properties
s Properties("MaxMBPerMin")=1000
s Status=##Class(DataMove.API).Generate(DMName,.Properties,.Warnings,.Errors)
i '$$$ISOK(Status) w !,$$SYSTEM.Status.GetErrorText(Status) q
#;At this point you could use the ^DATAMOVE utility to start, halt, monitor, and
#;activate the DataMove.
#;Or you can use the API's as follows:
#;Start the copy and dejournal processes
s Status=##Class(DataMove.API).StartCopy(Name)
i '$$$ISOK(Status) w !,$$SYSTEM.Status.GetErrorText(Status) q
#;Monitor state every 10 seconds
#;Activate will only run if the state is "Ready" and there is 20 MB or less
#;of journal to apply.
For {
  Hang 10
  Set Status=##Class(DataMove.API).GetProperties(DMName,.Properties)
  If (Properties("State")=$$$DMStateReady) {
    If (Properties("JRNMBToApply")<=20) {
      Quit
    }
  }
  Write !,"State:      "_Properties("ExpandedState")
  Write !,"MB to copy:  "_Properties("MBToCopy")
  Write !,"MB copied:  "_Properties("MBCopied")
  Write !,"MB checked: "_Properties("MBChecked")
  Write !,"Journal MB still to apply: "_Properties("JRNMBToApply")
}
w !,"DataMove is ready to Activate"
q

```