

## 記事

[Toshihiko Minamoto](#) · 2023年2月27日 7m read

## Ensembleでの孤立メッセージ

今回は、「孤立メッセージ」について説明します。

### 孤立メッセージとは何ですか？

すべてのメッセージボディは、メタデータを保持するメッセージヘッダと関連付けられます。ヘッダには、ソース構成名称、ターゲット構成名称、作成時刻、処理時刻、関連するメッセージボディ参照、セッション情報、メッセージボディのクラス名称、メッセージステータスなどの情報が格納されます。メッセージボディに対応するヘッダレコードが存在しない場合、そのメッセージボディは孤立メッセージボディと呼ばれます。ここでは、孤立メッセージボディの原因となる可能性があるものについて説明します。

### ヘッダーのみを削除する場合

削除タスクの設定において、BodiesToo  
メッセージヘッ

ダとともにメッセージボディも

削除するかどうかを指定するものです。この設定をOFF

にすると、削除タスクはメッセージヘッダーのみを削除し、メッセージボディは残します。これらのメッセージボディは、参照されたヘッダが削除されることから、孤立したレコードとなります。

メッセージヘッダの削除したが、メッセージボディは残している場合、マネジメントポータルでは孤立メッセージボディを削除する方法はありません。この場合、プログラムによってメッセージボディを削除する必要があります。

#### Task Scheduler Wizard

This wizard helps you schedule a task for execution by the Task Manager or to edit the details of a previously scheduled task. For user-defined tasks you must first create a new subclass of the %SYS.Task.Definition class which will then be selectable as a 'Task type'.

|  |   |
|--|---|
| Task name: *                           | <input type="text" value="PurgeMessages"/>  |
| Description:                           | <input type="text"/>  |
| Namespace to run task in:              | <input type="text" value="ENSEMBLE"/>   |
| Task type: *                           | <input type="text" value="Ens.Util.Tasks.Purge"/>   |
|  | <div><div>BodiesToo <input type="checkbox"/></div><div>KeepIntegrity <input checked="" type="checkbox"/></div><div>NumberOfDaysToKeep <input type="text" value="7"/></div><div>TypesToPurge <input type="text" value="Messages"/></div></div> |
| Task priority:                         | <input type="text" value="Priority Normal"/>  |
| Run task as this user:                 | <input type="text" value="UnknownUser"/>  |
| Open output file when task is running? | <input type="text" value="No"/>   |
| Output file:                           | <input type="text"/> <input type="button" value="Browse..."/>   |
| Suspend task on error?                 | <input type="text" value="No"/>   |
| Reschedule task after system restart?  | <input type="text" value="No"/>   |

削除タスクについては、ドキュメントをご参照ください。

<http://docs.intersystems.com/latest/csp/docbook/DocBook.UI.Page.cls?KEY=EGMGpurge#EGMGpurgebasic>

## 複合メッセージボディの クラス (オブジェクトの値によるプロパティ)

### Ensemble

がメッセージボディを削除する際、メッセージボディのオブジェクトの値によるプロパティが削除されるとは限りません。具体的には、他のオブジェクトを削除するのは、それらがシリアルオブジェクトであるか、子オブジェクト（関係によって定義される）である場合のみである。その他のオブジェクトについては、削除トリガーを定義するか、メッセージボディのクラスで %onDelete() 方法を実装して、適切に削除を処理する必要があります。

### onDelete実装のサンプルコード

```
Class Sample.Address Extends %Persistent{
  /// ???
  Property Street As %String(MAXLEN = 80);
  /// ???
  Property City As %String(MAXLEN = 80);
  /// 2????????
  Property State As %String(MAXLEN = 2);
  /// ???5??ZIP?Zone Improvement Plan????
  Property Zip As %String(MAXLEN = 5);
}
Class Sample.Person Extends %Persistent{
  /// ?????
  Property Name As %String [ Required ];
  /// ?????????????????????????????????
  Property SSN As %String(PATTERN = "3N1"-"2N1"-"4N") [ Required ];
  /// ?????
  Property DOB As %Date;
  /// ?????
  Property Home As Address;
  /// ?????
  Property Office As Address;
  ///????????????????????????
  ClassMethod %onDelete(oid As %ObjectIdentity) As %Status [ Private ]{
    // ?????????????????????????
    Set tSC = $$$OK, tThis = ##class(Sample.Person).%Open(oid)
    If $ISOBJECT(tThis.Home) Set tSC = ##class(Sample.Address).%DeleteId(tThis.Home.%Id())
    If $ISOBJECT(tThis.Office) Set tSC = ##class(Sample.Address).%DeleteId(tThis.Office.%Id())
    Quit tSC
  }
  ///SQL????????????/????
  Trigger onDelete [ Event = DELETE ]{
    // ?????????????????????????{%%ID} ????????????? ID ?????
    Set tID={%%ID}
    Set tThis = ##class(Sample.Person).%OpenId(tID)
    If $ISOBJECT(tThis.Home) Do ##class(Sample.Address).%DeleteId(tThis.Home.%Id())
    If $ISOBJECT(tThis.Office) Do ##class(Sample.Address).%DeleteId(tThis.Office.%Id())
    Quit
  }
}
```

## メッセージオブジェクトが作成されましたが、他のホストに送信されませんでした。

メッセージが別のホストに送信/転送されようとしたら、Ensembleは新しいメッセージヘッダを作成し、対応するメッセージボディを関連付けます。ビジネスサービス/プロセスで作成されたメッセージボディ/オブジェクトのインスタンスがディスク/データベースに保存され、かつプロダクション内の別のホストに送信されなかった場合、関連するヘッダーがなく、孤立メッセージボディとして残ります。ベストプラクティスは、転送されない限りメッセージボディを作成しないこと、またはオブジェクトのインスタンスの %Save() をコールしないことです (SendRequestSync/SendRequestAsync API は、メッセージをターゲット設定キューに入れる前にそれを保存します)。この方法では、メッセージボディのオブジェクトは、他のホストに送信されない限り、永続化されることはありません。

この問題の最も一般的な原因は、開発者が：

- メッセージボディを複製し、複製されたメッセージボディを転送しない場合です。
- コンテキスト変数 (BPL) で作成されたメッセージボディオブジェクトが転送されない場合です。

## 孤立メッセージの効果

孤立メッセージは、削除タスクによって削除されません。これらのメッセージはディスクスペースを消費し、その数が増えるにつれて、ディスク使用量も増加します。ディスク使用量は、メッセージボディのデータだけでなく、これらの孤立メッセージボディレコードのインデックス/サーチテーブルのエントリも含めて使用されることになります。

## 孤立したメッセージの特定

孤立したメッセージの存在は、メッセージヘッダとボディを照会することで特定することができます。各メッセージヘッダは、対応するメッセージボディを参照します。メッセージボディのオブジェクトIdの参照はヘッダの MessageBodyId プロパティに、メッセージボディのクラス名はヘッダの MessageBodyClassName プロパティに格納されます。

HL7 メッセージ・テーブルで孤立メッセージを見つけるためのクエリの例：

```
SELECT HL7.Id FROM EnsLibHL7.Message HL7
```

```
LEFT JOIN Ens.MessageHeader hdr
```

```
ON HL7.Id=hdr.MessageBodyId
```

```
WHERE hdr.MessageBodyId IS NULL
```

上記のクエリは、対応するヘッダを持たないすべての HL7 メッセージを返します。このクエリは、メッセージボディのテーブル名称を置き換えるだけで、他のメッセージタイプに対するクエリに変更することができます。

## 孤立メッセージの削除

マネジメントポータルでは孤立メッセージボディを削除する方法はありません。この場合、プログラムによってメッセージボディを削除する必要があります。ENSDEMO データベースでは、Demo.Util.CleanupSet というクラスが、これを実行する方法の例を示してくれます。このルーチンは、オブジェクトのプロパティ参照も含めて深く削除します。

孤立メッセージを削除するためのルーチンがもう一つありますが、このルーチンは深い削除はしませんし、メッセージボディの削除だけを助けます。ソースをダウンロードするために、以下にgithubへのリンクを載せておきます。

<https://gist.github.com/suriyasv/2ed7f2dbcf8c79f3b9938762c17c0b5>

ベストプラクティスは、

1. 孤立メッセージを防ぐために、常にプログラミングエラーを避けることです（前述）。
2. BodiesToo設定をOFFにした削除タスクを、ボディが必要な場合のみ設定し、これらのメッセージボディはプログラムによってのみ削除できることを認識することです。
3. 永続オブジェクトプロパティに対するRelationshipsまたはOnDelete実装することです。

この記事が、プロダクションを構築する際にお役に立てば幸いです。また、ご不明な点などございましたら、お知らせください。ありがとうございました。

[#システム管理](#) [#ヒントとコツ](#) [#ベストプラクティス](#) [#監視](#) [#相互運用性](#) [#Ensemble](#)

---

### ソースURL:

<https://jp.community.intersystems.com/post/ensemble%E3%81%A7%E3%81%AE%E5%AD%A4%E7%AB%8B%E3%83%A1%E3%83%83%E3%82%BB%E3%83%BC%E3%82%B8>