



インターシステムズ製品のプロ セス・メモリ

Version 2024.1
2024-06-03

インターシステムズ製品のプロセス・メモリ

InterSystems IRIS Data Platform Version 2024.1 2024-06-03

Copyright © 2024 InterSystems Corporation

All rights reserved.

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働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

インターシステムズ製品のプロセス・メモリ.....	1
1 概要	1
2 プロセス領域の管理	1
2.1 \$ZSTORAGE	1
2.2 プロセスあたりの最大メモリ。	2
2.3 \$STORAGE	2
3 <STORE> エラー	2
3.1 特別な考慮事項	3

インターシステムズ製品のプロセス・メモリ

1 概要

プロセスでは、目標を達成するために多くの異なるリソースを使用します。これらには、CPU サイクルの一部またはすべて、メモリ、外部ストレージ、ネットワーク帯域幅などが含まれます。このページでは、メモリ使用量について説明します。具体的には、以下のようなデータ・ストレージに InterSystems IRIS® が割り当てるメモリについて説明します。

- ・ パブリック変数とプライベート変数

これらは、値が最初に割り当てられるときに割り当てられるメモリ領域です。ローカル配列の場合、ローカル変数名とすべての添え字の値の組み合わせは、単一の変数値を参照します。

きわめて長い文字列が含まれる変数を除き、変数では [\\$STORAGE](#) に関連付けられている領域が使用されます。きわめて長い文字列が含まれる変数は異なる方法で格納され、[\\$STORAGE](#) に割り当てられている領域を使用することはありません。

- ・ オブジェクト・インスタンス

オブジェクトがインスタンス化されるたびに、参照先のオブジェクト(場合によっては複数のオブジェクト)の現在のコンテンツを保持するために領域が割り当てられます。その領域は、最後のオブジェクト参照が削除されると戻されます。

- ・ ローカル入出力バッファ

InterSystems IRIS は、プロセス領域内でこのプロセスによって使用されるデバイスに関連付けられている入出力バッファを格納します。詳細は、ObjectScript の [OPEN](#) コマンドおよび "[入出力デバイス・ガイド](#)" を参照してください。

2 プロセス領域の管理

プロセスは、上記のエンティティのために使用するメモリの初期プールで始まります。アプリケーションは、これらのエンティティを作成するときに上記プールからメモリを消費します。アプリケーションがこれらのエンティティを削除するとき、そのメモリはプールに返されます。例えば、ルーチンが実行を開始すると、ほとんどの場合一部のメモリを消費するローカル変数を作成します。ルーチンが戻り、これらの変数が範囲外になると、これらの変数によって使用されているメモリは返され、再使用できるようになります。

アプリケーションがメモリを必要としていて、要求を満たすために使用できる十分に大きい(連続した)メモリ領域がプロセスのプールにない場合、プロセスは、基礎となるオペレーティング・システムからの追加のメモリのまとまりを要求し、プールに追加します。その後、そのメモリのまとまりがまったく使用されなくなると、オペレーティング・システムに返されます。エンティティがメモリに割り当てられる順序とそれらのエンティティがメモリから削除される順序は相互にミラー・イメージではない場合があるため、実行が進行するにつれて、メモリはある程度断片化されます。これは、上記のオペレーティング・システムからのメモリの割り当ておよび割り当て解除に影響する可能性があります。

2.1 \$ZSTORAGE

InterSystems IRIS プロセスは、最大で 2TB のメモリを使用します。メモリ使用量の管理を支援するために、InterSystems IRIS は、メモリ消費の制限値をより小さく設定する方法を管理者またはアプリケーションに提供します。この値は各プロセ

スのシステム変数 **\$ZSTORAGE** に格納されるため、**\$ZSTORAGE** には常にプロセス・メモリの最大許容サイズ (KB 単位) が格納されます。

\$ZSTORAGE の値は、1KB 単位で指定されます。最小許容値は 256、つまり、256KB です。プロセスが **\$ZSTORAGE** に設定できる最大値は、メモリの 2TB ($2^{31} * 1KB$) です。値を、最小よりも小さく、または最大よりも大きくすると、それぞれ、最小または最大の既定値になります。

2.2 プロセスあたりの最大メモリ。

この値は管理ポータルで設定します。関連ページにアクセスするには、[システム管理]→[構成]→[システム構成]→[メモリと開始設定] を選択します。表示されるページで、[(KB)] に値を設定します。

構成ファイル (iris.cpf) で、このパラメータは **bbsiz** と呼ばれます。この値が、プロセス開始時の **\$ZSTORAGE** の初期値です。

注釈 プロセスのメモリ制限は、プロセスを ObjectScript の **JOB** コマンドで開始するときにも設定できます。

2.3 \$STORAGE

システム変数 **\$STORAGE** は、実行中プロセスで使用可能なストレージの量を表します。これはバイト単位で示されます。プロセスのメモリ要求が **\$STORAGE** の値よりも大きい場合、またはオペレーティング・システムからのメモリ割り当て要求が失敗した場合、<STORE> エラーが発生します。

3 <STORE> エラー

メモリに関するプロセスの要求を満たすと **\$STORAGE** の値が負の値になる場合、またはオペレーティング・システムからのメモリ割り当て要求が失敗した場合、<STORE> エラーが発生します。**\$STORAGE** が負の値になるときの <STORE> エラーの処理に関して、InterSystems IRIS プロセスは 2 つのモード (通常モードと低メモリ・モード) の 1 つになっていると考えられます。

- ・ 通常モード

プロセスが通常モードで、メモリの要求を行わないと **\$STORAGE** が負の値になるような要求を行うと、このプロセスは <STORE> エラーをスローし、低メモリ・モードになります。

- ・ 低メモリ・モード

低メモリ・モードでは、**\$STORAGE** を負の値にする操作が許可されます。これは、追加のメモリを許可し、アプリケーションがエラーを処理してクリーンアップできるようにするためです。**\$STORAGE** の値を 256KB 以上 (またはこれより小さい場合は **\$ZSTORAGE** の 25%) に増やせるように低メモリ・モードのプロセスがメモリを解放すると、プロセスは通常モードに戻ります。低メモリ・モードでは、約 1 MB の下限が **\$STORAGE** に設定されます。操作をしないと **\$STORAGE** がこの制限を下回ってしまうような操作を行うと <STORE> エラーになります。下限の値は、低メモリ・モードになると **\$STORAGE** の値マイナス 1 MB で定義されます。

注釈 プロセスは、**\$ZSTORAGE** を許容範囲の任意の値に設定できます。**\$ZSTORAGE** が現在使用中の値よりも小さい値に設定されると、**\$STORAGE** は負の値になります。プロセスが通常モードのときにこれが発生した場合、次にメモリを割り当てる操作を行うと、プロセスで <STORE> エラーが発生し、低メモリ・モードになり、下限値がその値マイナス 1 MB になります。プロセスが既に低メモリ・モードになっているときにこれが発生すると、下限値は変更されないままです。

-1 MB の低メモリ・モード制限を超えた結果発生した <STORE> エラーまたはオペレーティング・システムからのメモリの割り当てに失敗した結果発生した STORE エラーの場合、使用できるメモリが非常に少ないため、プロセスの動作は予

想できません。プロセスによってエラーを通常どおりに処理できる場合、エラー・ハンドラで〈STORE〉エラーが発生する場合、またはエラー・ハンドラを呼び出せず、プロセスが停止する場合があります。

エラー・ハンドラは、以下の方法の 1 つまたは複数を使用して〈STORE〉エラーを処理できます。

- ・ メモリ要求を引き起こした演算処理を中止し、〈STORE〉エラーが発生するまでに演算処理によって取得されたストレージがある場合、これを解放します。
- ・ 不必要なデータを除去することによって、利用可能なより多くのメモリを生成しようとします。
- ・ 開いているファイルを閉じるなど必要なクリーンアップを実行し、プログラムを終了します。
- ・ 今後のプロセスの続行およびより多くのメモリの要求を可能にする大きい値に `$ZSTORAGE` の値を設定します。

3.1 特別な考慮事項

3.1.1 プラットフォーム

ほとんどの InterSystems IRIS インスタンスは、InterSystems IRIS プロセスごとに割り当てられるメモリが 2TB 未満のシステム上で実行されます。そのようなシステム上では、InterSystems IRIS プロセスが利用可能なシステム・メモリ (実物理メモリと利用可能なスワップ領域の合計) を使い果たした場合、基盤となるシステムはいくつかの方法でこの状態に対処します。以下にいくつか例を挙げます。

- ・ 一部のプラットフォームでは、システムは InterSystems IRIS プロセスを終了させる信号を送信します。
- ・ 例えば Linux や AIX などのいくつかのプラットフォームでは、システムはヒューリスティック・アルゴリズムを使用して、もともとメモリを消費すると見なされるプロセスを強制終了します。これは InterSystems IRIS プロセスの場合もありますが、選択されたその他のプロセスの場合もあります。
- ・ いくつかのシステムでは、基盤となるオペレーティング・システムの実行をクラッシュするカーネル “パニック” を生成することによってメモリの枯渇に対処します。
- ・ システムの中には、メモリの枯渇状況に対処できるものがありますが、そのリカバリ結果は InterSystems IRIS プロセス内でアクセス違反となる場合があります。

プログラミング手法として、InterSystems IRIS プロセスが、基盤となるプラットフォームによって使用されるエラー・リカバリ・アルゴリズムに依存してしまうことはお勧めできません。代わりに、このようなプロセスは適切な設計、テストおよびログの記録を行って、プロセスがリソース要件を適切に予測および管理できるようにする必要があります。

