

SYBASE®

Historical Server ユーザーズ・ガイド

**Adaptive Server® Enterprise**

15.5

ドキュメント ID : DC36488-01-1550-01

改訂 : 2009 年 10 月

Copyright © 2010 by Sybase, Inc. All rights reserved.

このマニュアルは Sybase ソフトウェアの付属マニュアルであり、新しいマニュアルまたはテクニカル・ノートで特に示されないかぎりは、後続のリリースにも付属します。このマニュアルの内容は予告なしに変更されることがあります。このマニュアルに記載されているソフトウェアはライセンス契約に基づいて提供されるものであり、無断で使用することはできません。

このマニュアルの内容を弊社の書面による事前許可を得ずに、電子的、機械的、手作業、光学的、またはその他のいかなる手段によっても、複製、転載、翻訳することを禁じます。

#### マニュアルの注文

マニュアルの注文を承ります。ご希望の方は、サイベース株式会社営業部または代理店までご連絡ください。マニュアルの変更は、弊社の定期的なソフトウェア・リリース時にのみ提供されます。

Sybase の商標は、[Sybase trademarks ページ](http://www.sybase.com/detail?id=1011207) (<http://www.sybase.com/detail?id=1011207>) で確認できます。Sybase およびこのリストに掲載されている商標は、米国法人 Sybase, Inc. の商標です。® は、米国における登録商標であることを示します。

Java および Java 関連の商標は、米国およびその他の国における Sun Microsystems, Inc. の商標または登録商標です。

Unicode と Unicode のロゴは、Unicode, Inc. の登録商標です。

IBM および Tivoli は、International Business Machines Corporation の米国およびその他の国における登録商標です。

このマニュアルに記載されている上記以外の社名および製品名は、当該各社の商標または登録商標の場合があります。

Use, duplication, or disclosure by the government is subject to the restrictions set forth in subparagraph (c)(1)(ii) of DFARS 52.227-7013 for the DOD and as set forth in FAR 52.227-19(a)-(d) for civilian agencies.

Sybase, Inc., One Sybase Drive, Dublin, CA 94568.

# 目次

はじめに .....	ix
<b>第 1 章</b>	
<b>概要 .....</b>	<b>1</b>
Adaptive Server Enterprise Monitor の説明 .....	1
Adaptive Server Monitor のコンポーネント .....	1
Adaptive Server Enterprise Monitor のアーキテクチャ .....	2
Historical Server の概念 .....	3
<b>第 2 章</b>	
<b>Historical Server の設定 .....</b>	<b>9</b>
Historical Server 設定の概念 .....	9
Historical Server の制御ファイルとホーム・ディレクトリ .....	9
オペレーティング・システムの起動アカウント .....	10
Historical Server のスーパーユーザ・アカウント .....	11
Sybase Open Client/Server 接続 .....	11
UNIX プラットフォームでの初期設定 .....	12
UNIX プラットフォームの前提条件 .....	12
UNIX プラットフォームでの設定手順 .....	12
Windows プラットフォームでの初期設定 .....	15
Windows での前提条件 .....	15
Windows でのインストールの結果 .....	16
Windows での設定手順 .....	16
Historical Server 起動パラメータの設定 .....	20
関数 .....	20
構文 .....	20
パラメータ .....	20
Historical Server の複数インスタンスの設定 .....	23
Historical Server の複数インスタンスの作成時期 .....	23
UNIX プラットフォームでの追加 Historical Server の設定 .....	24
Windows プラットフォームでの追加 Historical Server の設定 .....	25

---

<b>第 3 章</b>	<b>Historical Server の起動と停止</b>	<b>33</b>
	UNIX プラットフォームでの Historical Server の起動と停止	33
	UNIX での Historical Server の起動	33
	UNIX での Historical Server の停止	34
	Windows での Historical Server の起動と停止	36
	Windows での Historical Server の起動	37
	レジストリからの起動パラメータの推定	37
	Historical Server が実行中であることの確認	37
	Windows での Historical Server の停止	38
<b>第 4 章</b>	<b>コマンド・リファレンス</b>	<b>41</b>
	コマンドの概要	41
	コマンドの構文	42
	コマンド・ステータスとコマンド・エラー	43
	Historical Server への入力手段としてのスクリプト・ファイル	44
	Historical Server への接続	44
	接続前の前提条件	44
	接続方法	45
	Historical Server のアクティビティに必要なパーミッション	45
	相互に排他的なセッション	46
	Historical Server のコマンド	48
	hs_create_alarm	48
	hs_create_filter	52
	hs_create_playback_session	55
	hs_create_playback_view	63
	hs_create_recording_session	65
	hs_create_view	68
	hs_delete_data	69
	hs_initiate_playback	70
	hs_initiate_recording	70
	hs_list	71
	hs_playback_sample	75
	hs_shutdown	79
	hs_status	81
	hs_terminate_playback	82
	hs_terminate_recording	82
<b>第 5 章</b>	<b>データ・ファイルと出力オプション</b>	<b>85</b>
	Historical Server データ・ファイルの概要	85
	Historical Server ファイルの説明	85
	ファイルのパーミッション	86
	一般的なファイル・フォーマット	86
	制御ファイル	86
	ヘッダ・レコード	87
	セッション制御レコード	88

---

ビュー制御レコード .....	89
データ項目制御レコード .....	90
アラーム制御レコード .....	90
フィルタ制御レコード .....	91
データ・ファイル .....	91
エラー・メッセージ・ファイル .....	92
スクリプト・ファイル .....	92
スクリプト・ファイル・テーブル名 .....	93
スクリプト・ファイルのテーブル・カラム名 .....	93
スクリプト・ファイル・コマンドの受け渡し .....	93
スクリプトの使用例 .....	94
バルク・コピーの例 .....	94
例 .....	95
Cut ユーティリティの例 .....	96
Historical Server でモニタリング・データのデータベースへの 出力を可能にする .....	97
受信側 Adaptive Server のセットアップ .....	98
Historical Server の起動 .....	98
データの表示 .....	99
Historical Server セッションの削除 .....	102
エラーのレポート .....	102
 付録 A    データ項目 .....	103
データ項目および定義テーブル .....	103
 付録 B    レコーディング・セッション・ビューの定義の仕様 .....	119
キーと結果の定義 .....	119
レコーディング・セッション・ビューの設計 .....	120
Process ID の使用 .....	120
アプリケーション名の使用 .....	121
ビューに空のローがある場合とローがない場合 .....	121
有効なキー・データ項目と結果データ項目の組み合わせ表 .....	122
有効な組み合わせの例 .....	144
無効な組み合わせの例 .....	144
各データ項目に対して有効な統計タイプの表 .....	145
 付録 C    プレイバック・ビューの定義の仕様 .....	153
要約レベルの詳細 .....	153
raw プレイバック .....	153
actual プレイバック .....	154
entire プレイバック .....	155
ユーザ定義の間隔でのプレイバック .....	155
要約インターバルのまとめ .....	156

---

プレイバック・ビューの設計.....	157
入力セッションの指定規則 .....	157
入力ビューとプレイバック・ビューの関係 .....	158
ビューの定義規則 .....	158
プレイバック・ビューのデータ項目要件の表 .....	160
データ項目に関する追加情報.....	165
“Timestamp”、“Timestamp Datim”、“Elapsed Time” の使用 .....	165
<b>付録 D レコーディング・セッション・ビューの例 .....</b>	<b>169</b>
キャッシュ・パフォーマンスの概要 .....	170
データベース・オブジェクトのロック・ステータス.....	170
データベース・オブジェクトのページ I/O .....	171
個別キャッシュに対するデータ・キャッシュ・アクティビティ .....	172
レコーディング・セッションにおけるデータ・キャッシュ統計 .....	172
サンプル・インターバルにおけるデータ・キャッシュ統計 .....	173
レコーディング・セッションにおけるデバイス I/O .....	173
サンプル間隔におけるデバイス I/O .....	174
デバイス I/O パフォーマンスの概要 .....	174
エンジン・アクティビティ .....	175
ロック・パフォーマンスの概要 .....	175
レコーディング・セッション中のネットワーク・アクティビティ .....	175
サンプル・インターバル中のネットワーク・アクティビティ .....	176
ネットワーク・パフォーマンスの概要 .....	176
ページ I/O .....	177
レコーディング・セッションにおけるプロシージャ・キャッシュ統計 .....	177
サンプル・インターバルにおけるプロシージャ・キャッシュ統計 .....	178
プロシージャ・ページ I/O .....	178
プロセス・アクティビティ .....	179
プロセスのデータベース・オブジェクト・ページ I/O .....	179
プロセスのロック詳細情報 .....	180
プロセスのページ I/O 詳細情報 .....	180
プロセスのロック情報 .....	181
プロセスのページ I/O 情報 .....	181
プロセス・ステータスの概要 .....	182
プロセスのストアド・プロシージャ・ページ I/O .....	182
サーバ・パフォーマンスの概要 .....	183
ストアド・プロシージャ・アクティビティ .....	183
トランザクション・アクティビティ .....	184
<b>索引 .....</b>	<b>185</b>





# はじめに

## 対象読者

このマニュアルは、次の読者を対象としています。

- Historical Server の設定および管理担当者
- Historical Server を使用して、Adaptive Server のパフォーマンスをモニタするユーザ

## このマニュアルの内容

このマニュアルの内容は、次のとおりです。

- 「[第1章 概要](#)」では、Adaptive Server® Enterprise Monitor の概要と Historical Server の基本概念について説明します。
- 「[第2章 Historical Server の設定](#)」では、UNIX と Windows の両プラットフォームについて Historical Server の初期設定を完了する方法と設定を変更する方法を説明します。
- 「[第3章 Historical Server の起動と停止](#)」では、UNIX と Windows の両プラットフォームでの Historical Server の起動方法と停止方法を説明します。
- 「[第4章 コマンド・リファレンス](#)」では、Historical Server の `isql` コマンド・インターフェースについて説明します。
- 「[第5章 データ・ファイルと出力オプション](#)」では、Historical Server によって作成されるファイルについて説明します。また、`bcp` ユーティリティを使用してデータを Adaptive Server テーブルにロードする方法など、これらのファイル内にあるデータへのアクセス方法についても説明します。
- 「[付録 A データ項目](#)」では、Historical Server から使用できるデータ項目について説明します。
- 「[付録 B レコーディング・セッション・ビューの定義の仕様](#)」では、レコーディング・セッション・ビューを定義するための、データ項目名と統計タイプの有効な組み合わせについて説明します。
- 「[付録 C プレイバック・ビューの定義の仕様](#)」では、プレイバック・セッション・ビューを定義するための、データ項目名と統計タイプの有効な組み合わせについて説明します。
- 「[付録 D レコーディング・セッション・ビューの例](#)」には、レコーディング・セッション・ビューの例が収録されています。

---

## 関連マニュアル

Adaptive Server® Enterprise には次のマニュアルが用意されています。必要に応じて参照してください。

- 使用しているプラットフォームの『リリース・ノート』 – マニュアルには記載できなかった最新の情報が記載されています。

このリリース・ノートの最新バージョン(英語版)を入手できます。製品のCDがリリースされた後で、製品またはマニュアルに関する重要な情報が追加されているかを確認するには、Sybase® Product Manuals Web サイトを使用してください。
- 使用しているプラットフォームの『インストール・ガイド』 – すべての Adaptive Server および関連する Sybase 製品のインストール、アップグレード、設定の手順について説明しています。
- 『新機能ガイド』 – Adaptive Server の新しい機能について説明しています。また、新しい機能をサポートするためのシステム変更や、既存のアプリケーションに影響を与える可能性がある変更についても説明しています。
- 『Active Messaging ユーザーズ・ガイド』 – Active Messaging を使用して、Adaptive Server Enterprise データベースでトランザクション(データ変更)を取得し、外部アプリケーションにイベントとしてリアルタイムで渡す方法について説明しています。
- 『コンポーネント統合サービス・ユーザーズ・ガイド』 – コンポーネント統合サービスを使用して、リモートの Sybase データベースおよび Sybase 以外のデータベースに接続する方法について説明しています。
- 使用しているプラットフォームの『設定ガイド』 – 特定の設定作業の手順について説明しています。
- 『用語解説』 – Adaptive Server マニュアルで使用されている技術用語について説明しています。
- 『Historical Server ユーザーズ・ガイド』 – Historical Server を使用して、Adaptive Server のパフォーマンス情報を入手する方法について説明しています。
- 『Adaptive Server Enterprise における Java』 – Adaptive Server データベースで Java クラスをデータ型、関数、ストアド・プロシージャとしてインストールして使用する方法について説明しています。
- 『Job Scheduler ユーザーズ・ガイド』 – コマンド・ラインまたはグラフィカル・ユーザ・インターフェース(GUI)を使用して、ローカルまたはリモートの Adaptive Server でジョブのインストール、設定、作成、スケジュールを行う方法について説明しています。
- 『マイグレーション技術ガイド』 – 別のバージョンの Adaptive Server にマイグレートするための方法とツールについて説明しています。

- ・ 『Monitor Client Library プログラマーズ・ガイド』 – Adaptive Server のパフォーマンス・データにアクセスする Monitor Client Library アプリケーションの記述方法について説明しています。
- ・ 『Monitor Server ユーザーズ・ガイド』 – Monitor Server を使用して、Adaptive Server のパフォーマンス統計を取得する方法について説明しています。
- ・ 『モニタリング・テーブル・ダイヤグラム』 – モニタリング・テーブルと、そのエンティティの関係をポスター形式で図解しています。フル・サイズのダイアグラムは印刷版だけで参照できます。コンパクト版は PDF 形式で参照できます。
- ・ 『パフォーマンス＆チューニング・シリーズ』 – Adaptive Server で最高のパフォーマンスを実現するためのチューニング方法について説明しています。
  - ・ 『基本』 – Adaptive Server のパフォーマンスに関する問題の理解と調査の基本について説明しています。
  - ・ 『統計的分析によるパフォーマンスの向上』 – Adaptive Server で統計情報がどのように保存され、表示されるかについて説明しています。また、`set statistics` コマンドを使用して、サーバの統計情報を分析する方法について説明しています。
  - ・ 『ロックと同時実行制御』 – ロック・スキームを使用してパフォーマンスを向上させる方法と、同時実行性を最小限に抑えるようにインデックスを選択する方法について説明しています。
  - ・ 『sp\_sysmon による Adaptive Server の監視』 – `sp_sysmon` を使用してパフォーマンスをモニタリングする方法について説明しています。
  - ・ 『モニタリング・テーブル』 – Adaptive Server のモニタリング・テーブルに統計情報や診断情報を問い合わせる方法について説明しています。
  - ・ 『物理データベースのチューニング』 – データの物理的配置、データに割り付けられた領域、テンポラリ・データベースの管理方法について説明しています。
  - ・ 『クエリ処理と抽象プラン』 – オプティマイザがクエリを処理する方法と、抽象プランを使用してオプティマイザのプランの一部を変更する方法について説明しています。
- ・ 『クイック・リファレンス・ガイド』 – コマンド、関数、システム・プロシージャ、拡張システム・プロシージャ、データ型、ユーティリティの名前と構文の包括的な一覧表を記載したポケット版 (PDF 版は通常サイズ) のマニュアルです。
- ・ 『リファレンス・マニュアル』 – 詳細な Transact-SQL® 情報を記載しています。

- 
- ・『ビルディング・ブロック』—データ型、関数、グローバル変数、式、識別子とワイルドカード、予約語について説明しています。
  - ・『コマンド』—コマンドについて説明しています。
  - ・『プロシージャ』—システム・プロシージャ、カタログ・ストアド・プロシージャ、システム拡張ストアド・プロシージャ、`dbcc`ストアド・プロシージャについて説明しています。
  - ・『テーブル』—システム・テーブル、モニタリング・テーブル、`dbcc`テーブルについて説明しています。
  - ・『システム管理ガイド』でさらに詳しく説明しています。
    - ・『第1巻』—設定パラメータ、リソースの問題、文字セット、ソート順、システムの問題の診断方法に関する説明を含め、システム管理の基本の概要について説明しています。『第1巻』の後半は、セキュリティ管理に関する詳細な説明です。
    - ・『第2巻』—物理的なリソースの管理、デバイスのミラーリング、メモリとデータ・キャッシュの設定、マルチプロセッサ・サーバとユーザ・データベースの管理、データベースのマウントとマウント解除、セグメントの作成と使用、`reorg`コマンドの使用、データベース一貫性の検査方法についての手順とガイドラインを説明しています。『第2巻』の後半では、システムとユーザ・データベースをバックアップおよびリストアする方法について説明しています。
  - ・『システム・テーブル・ダイアグラム』—システム・テーブルと、そのエンティティとの関係をポスター形式で図解しています。フル・サイズのダイアグラムは印刷版だけで参照できます。コンパクト版はPDF形式で参照できます。
  - ・『Transact-SQL ユーザーズ・ガイド』—リレーショナル・データベース言語の拡張版である Sybase の Transact-SQL について説明しています。まだ経験の浅いデータベース管理システムのユーザは、このマニュアルをガイドブックとして使用してください。`pubs2` および `pubs3` サンプル・データベースの詳細も説明しています。
  - ・『トラブルシューティング・シリーズ』—
    - ・トラブルシューティング『エラー・メッセージと詳細な解決方法』—発生する可能性のある問題について、トラブルシューティング手順を説明しています。このマニュアルで取り上げられている問題は、Sybase 製品の保守契約を結んでいるサポート・センタに最も頻繁に寄せられるものです。
    - ・『トラブルシューティング&エラー・メッセージ・ガイド』—発生頻度が高い Adaptive Server のエラー・メッセージの解決方法について詳しい手順を説明しています。
  - ・『暗号化カラム・ユーザーズ・ガイド』—Adaptive Server を使用して暗号化カラムを設定し、使用する方法について説明しています。

- ・『インメモリ・データベース・ユーザーズ・ガイド』－ インメモリ・データベースの設定および使用方法について説明しています。
- ・『Adaptive Server 分散トランザクション管理機能の使用』－ 分散トランザクション処理環境での Adaptive Server DTM 機能の設定、使用、トラブルシューティングについて説明しています。
- ・『IBM® Tivoli® Storage Manager と Backup Server の使用』－ IBM Tivoli Storage Manager を設定および使用して Adaptive Server のバックアップを作成する方法について説明しています。
- ・『高可用性システムにおける Sybase フェールオーバの使用』－ Sybase のフェールオーバ機能を使用して、Adaptive Server を高可用性システムのコンパニオン・サーバとして設定する方法について説明しています。
- ・『Unified Agent および Agent Management Console』－ Unified Agent について説明しています。Unified Agent は、分散 Sybase リソースを管理、モニタ、制御するためのランタイム・サービスを提供します。
- ・『ユーティリティ・ガイド』－ オペレーティング・システム・レベルで実行される `isql` および `bcp` などの、Adaptive Server のユーティリティ・プログラムについて説明しています。
- ・『Web Services ユーザーズ・ガイド』－ Adaptive Server 用の Web サービスの設定、使用、トラブルシューティング方法について説明しています。
- ・『XA インタフェース統合ガイド for CICS, Encina, TUXEDO』－ X/Open XA トランザクション・マネージャを備えた Sybase DTM XA インタフェースを使用する方法について説明しています。
- ・『Adaptive Server Enterprise における XML サービス』－ データベースに XML 機能を導入する、Sybase ネイティブの XML プロセッサと Sybase Java ベースの XML のサポートについて、また XML サービスに準拠したクエリとマッピング用の関数について説明しています。

#### その他の情報

Sybase Getting Started CD、SyBooks™ CD、Sybase® Product Manuals Web サイトを利用すると、製品について詳しく知ることができます。

- ・ Getting Started CD には、PDF 形式のリリース・ノートとインストール・ガイド、SyBooks CD に含まれていない他のマニュアルや更新情報が収録されています。この CD は製品のソフトウェアに同梱されています。Getting Started CD に収録されているマニュアルを参照または印刷するには、Adobe Acrobat Reader が必要です (CD 内のリンクを使用して Adobe の Web サイトから無料でダウンロードできます)。
  - ・ SyBooks CD には製品マニュアルが収録されています。この CD は製品のソフトウェアに同梱されています。Eclipse ベースの SyBooks ブラウザを使用すれば、使いやすい HTML 形式のマニュアルにアクセスできます。
- 一部のマニュアルは PDF 形式で提供されています。これらのマニュアルは SyBooks CD の PDF ディレクトリに収録されています。PDF ファイルを開いたり印刷したりするには、Adobe Acrobat Reader が必要です。

---

SyBooks をインストールして起動するまでの手順については、Getting Started CD の『SyBooks インストール・ガイド』、または SyBooks CD の *README.txt* ファイルを参照してください。

- Sybase Product Manuals Web サイトは、SyBooks CD のオンライン版であり、標準の Web ブラウザを使用してアクセスできます。また、製品マニュアルのほか、EBFs/Updates、Technical Documents、Case Management、Solved Cases、ニュース・グループ、Sybase Developer Network へのリンクもあります。

Technical Library Product Manuals Web サイトにアクセスするには、Product Manuals (<http://www.sybase.com/support/manuals/>) にアクセスしてください。

**Web 上の Sybase 製品の動作確認情報**

Sybase Web サイトの技術的な資料は頻繁に更新されます。

❖ **製品認定の最新情報にアクセスする**

- 1 Web ブラウザで Technical Documents を指定します。  
(<http://www.sybase.com/support/techdocs/>)
- 2 [Certification Report] をクリックします。
- 3 [Certification Report] フィルタで製品、プラットフォーム、時間枠を指定して [Go] をクリックします。
- 4 [Certification Report] のタイトルをクリックして、レポートを表示します。

❖ **コンポーネント認定の最新情報にアクセスする**

- 1 Web ブラウザで Availability and Certification Reports を指定します。  
(<http://certification.sybase.com/>)
- 2 [Search By Base Product] で製品ファミリとベース製品を選択するか、[Search by Platform] でプラットフォームとベース製品を選択します。
- 3 [Search] をクリックして、入手状況と認定レポートを表示します。

❖ **Sybase Web サイト (サポート・ページを含む) の自分専用のビューを作成する**

MySybase プロファイルを設定します。MySybase は無料サービスです。このサービスを使用すると、Sybase Web ページの表示方法を自分専用にカスタマイズできます。

- 1 Web ブラウザで Technical Documents を指定します。  
(<http://www.sybase.com/support/techdocs/>)
- 2 [MySybase] をクリックし、MySybase プロファイルを作成します。

- ❖ **製品認定の最新情報にアクセスする**
  - 1 Web ブラウザで Technical Documents を指定します。  
(<http://www.sybase.com/support/techdocs/>)
  - 2 [Certification Report] をクリックします。
  - 3 [Certification Report] フィルタで製品、プラットフォーム、時間枠を指定して [Go] をクリックします。
  - 4 [Certification Report] のタイトルをクリックして、レポートを表示します。
- ❖ **コンポーネント認定の最新情報にアクセスする**
  - 1 Web ブラウザで Availability and Certification Reports を指定します。  
(<http://certification.sybase.com/>)
  - 2 [Search By Base Product] で製品ファミリとベース製品を選択するか、[Search by Platform] でプラットフォームとベース製品を選択します。.
  - 3 [Search] をクリックして、入手状況と認定レポートを表示します。
- ❖ **Sybase Web サイト (サポート・ページを含む) の自分専用のビューを作成する**

MySybase プロファイルを設定します。MySybase は無料サービスです。このサービスを使用すると、Sybase Web ページの表示方法を自分専用にカスタマイズできます。

  - 1 Web ブラウザで Technical Documents を指定します。  
(<http://www.sybase.com/support/techdocs/>)
  - 2 [MySybase] をクリックし、MySybase プロファイルを作成します。

## Sybase EBF とソフトウェア・メンテナンス

- ❖ **EBF とソフトウェア・メンテナンスの最新情報にアクセスする**
  - 1 Web ブラウザで Sybase Support Page を指定します。  
(<http://www.sybase.com/support>)
  - 2 [EBFs/Maintenance] を選択します。MySybase のユーザ名とパスワードを入力します。
  - 3 製品を選択します。
  - 4 時間枠を指定して [Go] をクリックします。EBF/Maintenance リリースの一覧が表示されます。

鍵のアイコンは、「Technical Support Contact」として登録されていないため、一部の EBF/Maintenance リリースをダウンロードする権限がないことを示しています。未登録でも、Sybase 担当者またはサポート・コンタクトから有効な情報を得ている場合は、[Edit Roles] をクリックして、「Technical Support Contact」の役割を MySybase プロファイルに追加します。

- 
- 5 EBF/Maintenance レポートを表示するには [Info] アイコンをクリックします。ソフトウェアをダウンロードするには製品の説明をクリックします。

## 表記規則

次の項では、このマニュアルで使用されている表記について説明します。

SQL は自由な形式の言語で、1 行内のワード数や、改行の仕方に規則はありません。このマニュアルでは、読みやすくするため、例や構文を文の句ごとに改行しています。複数の部分からなり、2 行以上にわたる場合は、字下げしています。複雑なコマンドの書式には、修正された BNF (Backus Naur Form) 記法が使用されています。

**表 1** に構文の規則を示します。

**表 1: このマニュアルでのフォントと構文規則**

要素	例
コマンド名、プロシージャ名、ユーティリティ名、その他他のキーワードは sans serif フォントで表記する。	<code>select</code> <code>sp_configure</code>
データベース名とデータ型は sans serif フォントで表記する。	<code>master</code> データベース
ファイル名、変数、パス名は斜体で表記する。	システム管理ガイド <code>sql.ini</code> ファイル <code>column_name</code> \$SYBASE/ASE ディレクトリ
変数 ( ユーザが入力する値を表す語 ) がクエリまたは文の一部である場合は Courier フォントの斜体で表記する。	<code>select column_name</code> <code>from table_name</code> <code>where search_conditions</code>
カッコはコマンドの一部として入力する。	<code>compute row_aggregate (column_name)</code>
2 つのコロンと等号は、構文が BNF 表記で記述されていることを示す。この記号は入力しない。「～と定義されている」ことを意味する。	<code>::=</code>
中カッコは、その中のオプションを 1 つ以上選択しなければならないことを意味する。コマンドには中カッコは入力しない。	<code>{cash, check, credit}</code>
角カッコは、オプションを選択しても省略してもよいことを意味する。コマンドには角カッコは入力しない。	<code>[cash   check   credit]</code>
中カッコまたは角カッコの中のカンマで区切られたオプションをいくつでも選択できることを意味する。複数のオプションを選択する場合には、オプションをカンマで区切る。	<code>cash, check, credit</code>
パイプまたは縦線は複数のオプションのうち 1 つだけを選択できることを意味する。	<code>cash   check   credit</code>

---

要素	例
省略記号 (...) は、直前の要素を必要な回数だけ繰り返し指定できることを意味する。	<pre>buy thing = price [cash   check   credit] [, thing = price [cash   check   credit]]...</pre> <p>この例では、製品 (thing) を少なくとも 1 つ購入 (buy) し、価格 (price) を指定する必要があります。支払方法を選択できる。角カッコで囲まれた項目の 1 つを選択する。追加品目を、必要な数だけ購入することもできる。各 buy に対して、購入した製品 (thing)、価格 (price)、オプションで支払方法 (cash, check, credit のいずれか) を指定します。</p>

---

- 次は、オプション句のあるコマンドの構文の例です。

```
sp_dropdevice [device_name]
```

複数のオプションを持つコマンドの例を示します。

```
select column_name
  from table_name
 where search_conditions
```

構文では、キーワード (コマンド) は通常のフォントで表記し、識別子は小文字で表記します。ユーザが提供するワードは斜体で表記します。

- Transact-SQL コマンドの使用例は次のように表記します。

```
select * from publishers
```

- 次は、コンピュータからの出力例です。

pub_id	pub_name	city	state
0736	New Age Books	Boston	MA
0877	Binnet & Hardley	Washington	DC
1389	Algodata Infosystems	Berkeley	CA

(3 rows affected)

このマニュアルでは、例に使用する文字はほとんどが小文字ですが、Transact-SQL のキーワードを入力するときは、大文字と小文字は区別されません。たとえば、SELECT、Select、select はすべて同じです。

テーブル名などのデータベース・オブジェクトの大文字と小文字を Adaptive Server が区別するかどうかは、Adaptive Server にインストールされたソート順によって決まります。シングルバイト文字セットを使用している場合は、Adaptive Server のソート順を再設定することによって、大文字と小文字の区別の取り扱い方を変更できます。詳細については、『システム管理ガイド』を参照してください。

---

## アクセシビリティ機能

このマニュアルには、アクセシビリティを重視した HTML 版もあります。この HTML 版マニュアルは、スクリーン・リーダーで読み上げる、または画面を拡大表示するなどの方法により、その内容を理解できるよう配慮されています。

Adaptive Server HTML マニュアルは、連邦リハビリテーション法第 508 条のアクセシビリティ規定に準拠していることがテストにより確認されています。第 508 条に準拠しているマニュアルは通常、World Wide Web Consortium (W3C) の Web サイト用ガイドラインなど、米国以外のアクセシビリティ・ガイドラインにも準拠しています。

---

**注意** アクセシビリティ・ツールを効率的に使用するには、設定が必要な場合もあります。一部のスクリーン・リーダーは、テキストの大文字と小文字を区別して発音します。たとえば、すべて大文字のテキスト (ALL UPPERCASE TEXT など) はイニシャルで発音し、大文字と小文字の混在したテキスト (Mixed Case Text など) は単語として発音します。構文規則を発音するようにツールを設定すると便利かもしれません。詳細については、ツールのマニュアルを参照してください。

---

Sybase のアクセシビリティに対する取り組みについては、**Sybase Accessibility** (<http://www.sybase.com/accessibility>) を参照してください。Sybase Accessibility サイトには、第 508 条と W3C 標準に関する情報へのリンクもあります。

## 不明な点があるときは

Sybase ソフトウェアがインストールされているサイトには、Sybase 製品の保守契約を結んでいるサポート・センタとの連絡担当の方 (コンタクト・パーソン) を決めてあります。マニュアルだけでは解決できない問題があった場合には、担当の方を通して Sybase のサポート・センタまでご連絡ください。

## 概要

トピック名	ページ
<a href="#">Adaptive Server Enterprise Monitor の説明</a>	1
<a href="#">Adaptive Server Enterprise Monitor のアーキテクチャ</a>	3

## Adaptive Server Enterprise Monitor の説明

Adaptive Server Enterprise Monitor™ (Adaptive Server Monitor) では、Adaptive Server のパフォーマンスをリアルタイムまたは履歴データ収集モードでモニタできます。システム管理者は、この情報を使って潜在的なリソースのボトルネックを調べ、現在起きている問題を調査して、パフォーマンスを向上させるためにチューニングすることができます。Adaptive Server Monitor は、チューニングのためのフィードバックを次のレベルで提供します。

- Adaptive Server の設定
- データベースの設計
- アプリケーションとストアド・プロシージャの SQL 文

## Adaptive Server Monitor のコンポーネント

Adaptive Server Monitor は、Adaptive Server パフォーマンス・データを収集または表示する次の 4 つのコンポーネントから構成されています。

- Adaptive Server Enterprise Monitor Server (Monitor Server) – このサーバは、Adaptive Server のパフォーマンス・データをリアルタイムで収集し、そのデータを他の Adaptive Server Monitor のコンポーネントに利用できるようにします。Monitor Server は、Sybase Open Server アプリケーションです。
- Adaptive Server Enterprise Monitor Historical Server (Historical Server) – このサーバは、Adaptive Server のパフォーマンス・データを Monitor Server から取得し、そのデータをファイルに保存して後から分析できるようにします。Historical Server は、Sybase Open Server アプリケーションです。

- Adaptive Server Enterprise Plug-in for Sybase Central™ のモニタ (Monitor Viewer) – このモニタは、Adaptive Server のパフォーマンス・データを Monitor Server から取得し、そのデータをリアルタイムで表とグラフを使用して表示します。
- Adaptive Server Enterprise Monitor Client Library (Monitor Client Library) – Monitor Server と Historical Server に対するこのアプリケーション・プログラミング・インターフェース (API) は、ユーザがモニタリング・アプリケーションを作成するのに利用できます。Historical Server と、Adaptive Server Enterprise Plug-in for Sybase Central のモニタは、Monitor Client Library アプリケーションです。

## Adaptive Server Enterprise Monitor のアーキテクチャ

Adaptive Server は、Monitor Server が読み込む共有メモリ領域にパフォーマンス・データを保存します。この共有メモリ技術が採用されているため、Monitor Server は、モニタされている Adaptive Server と同じマシンにインストールし、実行する必要があります。Adaptive Server と Monitor Server との間には 1 対 1 の関係が成り立ちます。Monitor Server の詳細については、『ASE Monitor Server ユーザーズ・ガイド』を参照してください。

Monitor Client Library アプリケーションは、Adaptive Server のパフォーマンス統計を Monitor Server から取得します。これらのアプリケーションは、Monitor Server のクライアントです。パフォーマンス上の理由から、Monitor Client Library アプリケーションは、Adaptive Server と Monitor Server のペアが動作しているマシンとは別のマシンで実行することをおすすめします。詳細については、『ASE Monitor Client Library プログラマーズ・ガイド』を参照してください。

Adaptive Server Enterprise Plug-in for Sybase Central は、Adaptive Server のリソースの使用状況をさまざまな角度から、またさまざまな詳細レベルで表示するモニタのセットで構成されています。各オーブン・モニタは個別のアプリケーションで、Monitor Server とユニークなクライアント接続でつながっています。Sybase Central では、各 Adaptive Server インストール環境に、モニタ・オブジェクトのセットが保存されている [Monitors] フォルダが 1 つずつあります。

Historical Server は、Monitor Server からパフォーマンス情報を収集し、この情報を後から分析できるようにファイルに保存します。ユーザは Historical Server インタフェースを使用して、収集するデータと希望する収集期間を指定できます。インターフェースには、履歴データのプレイバック機能もあります。これらのインターフェースを次に示します。

- `isql` によるコマンド・インターフェース。詳細については、『ASE Monitor Historical Server ユーザーズ・ガイド』を参照してください。
- Monitor Client Library をを使ったプログラミング・インターフェース。詳細については、『ASE Monitor Client Library プログラマーズ・ガイド』を参照してください。

## Historical Server の概念

この項では、以下の Historical Server の概念について説明します。

- レコーディング・セッション
- プレイバック・セッション
- ビュー
- データ項目と統計タイプ

### レコーディング・セッション

レコーディング・セッションでは、Adaptive Server のパフォーマンス・データを収集し、後で分析できるようにファイルに格納します。レコーディング・セッションには以下の属性があります。

- Monitor Server の名前 – 名前と対応させてパフォーマンスをモニタする Adaptive Server を定義する。
- サンプル・インターバル – この属性でパフォーマンス・データを収集する頻度を定義する。
- ビュー、アラーム、フィルター – ビューとフィルタによって、収集するデータを定義する。アラームでは、特定のデータ項目が定義済みのスレッシュホールド値に達したときに実行する動作を定義する。
- 開始時刻と終了時刻 – これらを指定してデータを収集する時間を定義する。

レコーディング・セッションを作成するには、Historical Server の `isql` コマンド・インターフェース内で以下の一連のコマンドを使用します。

- `hs_create_recording_session`
- `hs_create_view`
- `hs_create_filter` (オプション)
- `hs_create_alarm` (オプション)
- `hs_initiate_recording`

レコーディング・セッションを作成すると、Historical Server によってセッション ID が割り当てられます。`hs_list` コマンドを使用して、定義したレコーディング・セッションのセッション ID をリストできます。`hs_list` コマンドでは、ビューノード名およびビュー内のデータ項目、アラーム、フィルタなど、レコーディング・セッションの完全な定義も表示できます。

Historical Server は、これらのレコーディング・セッションの定義を制御ファイルに格納します。この制御ファイルは、Historical Server のホーム・ディレクトリにあります。したがって、`hs_list` を使用して表示できるのは、現在の Historical Server と同じホーム・ディレクトリを使用している Historical Server インスタンスによって作成されたレコーディング・セッションの定義だけになります。Historical Server ホーム・ディレクトリの設定の詳細については、「[Historical Server の複数インスタンスの設定](#)」(23 ページ) を参照してください。

レコーディング・セッションで収集されたデータを調べるには、以下の方法があります。

- Sybase バルク・コピー (`bcp`) ユーティリティを使用して、Adaptive Server テーブルにレコーディング・セッションのデータを移植する方法。詳細については、「[バルク・コピーの例](#)」(94 ページ) を参照してください。
- Historical Server のプレイバック・セッションを開始する方法。

## プレイバック・セッション

プレイバック・セッションでは、1 つ以上のレコーディング・セッションの間に収集されたデータを取り出せます。データは以下の 2 つの方法でプレイバックできます。

- クライアントへのプレイバック – プレイバックの結果はユーザに送信される。ユーザは結果を端末に表示したり、ファイルにリダイレクトしたりできる。
- ファイルへのプレイバック – プレイバックの結果はファイルに格納される。作成されるファイルは、本質的に新しいレコーディング・セッションになる。これらのファイルは、別のプレイバック・セッションへの入力として使用したり、`bcp` ユーティリティへの入力として使用して Adaptive Server テーブルに移植したりできる。また、レコーディング・セッション・ファイルと同じように使用できる。

次の属性を使用して、プレイバック・セッションを定義します。

- 入力レコーディング・セッション – プレイバック・セッションへの入力は、1 つ以上のレコーディング・セッションになる。
- ビュー、開始時刻、終了時刻 – これらの属性を使用して、プレイバック・セッションに含める入力レコーディング・セッションからのデータを定義する。
- 要約レベル – 記録された内容を正確に表示するロー・プレイバックまたは他のさまざまな要約レベルを指定できる。

プレイバック・セッションを作成するには、Historical Server のコマンド・インターフェースで以下の一連のコマンドを使用します。

- [hs\\_create\\_playback\\_session](#)
- [hs\\_create\\_playback\\_view](#)
- [hs\\_initiate\\_playback](#)
- [hs\\_playback\\_sample](#) (クライアントへのプレイバックにのみ使用)
- [hs\\_terminate\\_playback](#)

## ビュー

レコーディング・セッション・ビューでは、Historical Server によって収集するパフォーマンス・データを定義します。プレイバック・セッション・ビューでは、レコーディング・セッション・ビューのどのパフォーマンス・データを Historical Server によってプレイバックするか定義します。

ビューは、ビューノミーと 1 つ以上のデータ項目で構成されています。各データには対応する統計タイプがあります。詳細については、「[データ項目と統計タイプ](#) (6 ページ)」を参照してください。

レコーディング・セッションを定義する場合、そのレコーディング・セッションに含める 1 つ以上のビューを定義します。レコーディング・セッションには、少なくとも 1 つのビューが必要です。レコーディング・セッション・ビューの作成の詳細については、「[hs\\_create\\_view \(68 ページ\)](#)」を参照してください。

「[付録 D レコーディング・セッション・ビューの例](#)」には、多数のレコーディング・セッション・ビューの例が収録されています。

プレイバック・セッションを定義する場合は、定義済みのレコーディング・セッションの中にあるどのビューをプレイバック・セッションに含めるか定義します。プレイバック・セッション・ビューには、レコーディング・セッション・ビューと同じ名前を付ける必要があります。レコーディング・セッション・ビューのすべてのデータ項目またはデータ項目のサブセットを、対応するプレイバック・ビューに含ませることができます。プレイバック・ビューの作成の詳細については、「[hs\\_create\\_playback\\_view \(63 ページ\)](#)」を参照してください。

## データ項目と統計タイプ

「データ項目」で、ビューに表示する特定の情報を指定します。データ項目に埋め込みスペースが含まれている場合、使用するときにはその名前を引用符で囲みます。データ項目には、ページ I/O、ログイン名、CPU 時間などがあります。

[表 A-1 \(103 ページ\)](#) に、使用可能なデータ項目をすべてリストし、それぞれの項目について説明します。

各データ項目には、「統計タイプ」があります。統計タイプでデータ項目の期間(サンプルまたはセッション)を定義し、またデータ項目に対して Historical Server が計算を実行するかも定義します。

統計タイプには埋め込みスペースが含まれています。統計タイプを Historical Server コマンドで使用するときは、引用符で囲む必要があります。

データ項目によっては、無効な統計タイプもあります。[表 B-3 \(145 ページ\)](#) で、各データ項目に対して有効な統計タイプを示します。

統計タイプは次の 6 つです。

- “Value for Sample” – この統計タイプは、アクティビティの数または最後のサンプル・インターバルに適用された情報のタイプを返す。計算は実行されない。
- アクティビティの数 – アクティビティの数を表すデータ項目では、この統計タイプによって最後のサンプル・インターバルで発生したアクティビティの回数を返す。たとえば、ページ I/O の Value for Sample は、最後のサンプル間隔で発生したページ I/O 数になる。
- その他の情報 – 文字列を表すデータ項目に対して有効な唯一の統計タイプ。たとえば、オブジェクト名の Value for Sample は、テーブルの名前を返す。この統計タイプは、ID や設定済みパラメータの値 (Process ID や Code Memory Size など) を表すデータ項目の唯一有効な統計タイプでもある。
- “Value for Session” – この統計タイプは、データの収集を開始してから(接続を開いてから)数えたアクティビティの累積数を返す。計算は実行されない。たとえば、Page I/O の Value for Session は、レコーディング・セッションが開始されてから発生したページ I/O 数になる。
- “Rate for Sample” – この統計タイプは、1 秒当たりの比率を計算する。最後のサンプル・インターバルにおけるアクティビティの 1 秒当たりの平均発生回数を返す。たとえば、ページ I/O の Rate for Sample は、最後のサンプル間隔で 1 秒ごとに発生したページ I/O の平均回数になる。

計算式は次のとおり。

最後のサンプル・インターバルでの回数

-----  
サンプル・インターバルの秒数

- “Rate for Session” – この統計タイプは、1秒当たりの比率を計算する。現在のレコーディング・セッションにおけるアクティビティの1秒当たりの平均発生回数を返す。たとえば、ページI/OのRate for Sessionは、レコーディング・セッションを開始してから1秒ごとに発生したページI/Oの平均回数になる。

計算式は次のとおり。

$$\frac{\text{セッションでの回数}}{\text{セッションの秒数}}$$

- “Average for Sample” – この統計タイプは、最後のサンプル間隔に発生したアクティビティ1回当たりの平均値を計算する。少数のデータ項目だけがこの統計タイプを使用できる。戻り値の意味は、データ項目によって異なる。たとえば、Procedure Elapsed TimeのAverage for Sampleは、最後のサンプル・インターバルにおいてストアド・プロシージャの実行1回当たりにかかった平均時間になる。
- “Average for Session” – この統計タイプは、セッションで発生したアクティビティ1回当たりの平均値を計算する。少数のデータ項目だけがこの統計タイプを使用できる。戻り値の意味は、データ項目によって異なる。たとえば、Procedure Elapsed TimeのAverage for Sessionは、レコーディング・セッションにおいてストアド・プロシージャの実行1回当たりにかかった平均時間になる。

表B-3(145ページ)で、データ項目と統計タイプの有効な組み合わせを示します。



トピック名	ページ
<a href="#">Historical Server 設定の概念</a>	<a href="#">9</a>
<a href="#">UNIX プラットフォームでの初期設定</a>	<a href="#">12</a>
<a href="#">Windows プラットフォームでの初期設定</a>	<a href="#">15</a>
<a href="#">Historical Server 起動パラメータの設定</a>	<a href="#">20</a>
<a href="#">Historical Server の複数インスタンスの設定</a>	<a href="#">23</a>

## Historical Server 設定の概念

この項では、Historical Server を設定する前に理解する必要のある次の概念について説明します。

- [Historical Server の制御ファイルとホーム・ディレクトリ](#)
- [オペレーティング・システムの起動アカウント](#)
- [Historical Server のスーパーユーザ・アカウント](#)

### Historical Server の制御ファイルとホーム・ディレクトリ

Historical Server の制御ファイルには、ユーザが作成するレコーディング・セッションに関する情報が保存されます。この情報は、再起動しても保持されるため、ユーザは以前に Historical Server を実行したときに作成したレコーディング・セッションにアクセスできます。制御ファイルによって、ユーザによるプライベート・レコーディング・セッション・ファイルへのアクセスが制限されます(レコーディング・セッション・ファイルへのアクセス方法は、パブリックとプライベートの 2 つがあります)。

Historical Server ホーム・ディレクトリが重要である理由として、次の 2 つが挙げられます。

- ホーム・ディレクトリには、Historical Server 制御ファイルが格納されます。Historical Server を起動すると、ホーム・ディレクトリにある制御ファイルが検索されます。制御ファイルがない場合は、Historical Server によって作成されます。

- ホーム・ディレクトリは、レコーディング・セッション中に Historical Server が書き込むデータ・ファイルのデフォルト・ディレクトリです。レコーディング・セッションを作成するユーザは、このデフォルトのロケーションを上書きできます。

Historical Server 起動コマンドの `-D` パラメータを使用して、ホーム・ディレクトリのロケーションを指定します。このパラメータは必須です。

以前に実行したときと同じホーム・ディレクトリにある同一の制御ファイルを使用している場合のみ、Historical Server の実行によって、前回の実行に使用したデータ・ファイルにアクセスできます。このため、ほとんどの場合、2つ以上の Historical Server を起動したとき、その Historical Server のホーム・ディレクトリを変更しないでください。異なるホーム・ディレクトリの使用については、[「Historical Server の複数インスタンスの設定」\(23 ページ\)](#) を参照してください。

Historical Server によって作成されるファイルの詳細については、[「第 5 章 データ・ファイルと出力オプション」](#) を参照してください。

### 制御ファイル情報へのアクセス

Historical Server の `hs_list` コマンドを使用して、Historical Server 制御ファイルの情報にアクセスできます。

### 制御ファイルの編集の禁止

制御ファイルを編集しないでください。ファイルが壊れることがあります。使用しているエディタに関係なく、制御ファイルを開いて、保存しないでください。

Historical Server を Windows で実行する場合には特に注意が必要です。Historical Server によって作成される他のファイルと異なり、制御ファイルは Windows の標準フォーマットのテキスト・ファイルではありません。制御ファイルの各テキスト行の行末には改行文字だけしかなく、通常の復帰／改行文字は使用しません。編集プログラムを使用すると、テキストに不要な復帰／改行文字が埋め込まれ、ファイルが壊れてしまう可能性があります。

### オペレーティング・システムの起動アカウント

Historical Server を起動するアカウントは、次の条件を満たしている必要があります。

- 毎回、同じアカウントで Historical Server を起動すること。

Historical Server が制御ファイルを作成するとき、Historical Server を初めて起動するオペレーティング・システム・アカウントが、それ以降その制御ファイルにアクセスできる唯一のアカウントとなります。制御ファイルにアクセスするには、それ以降も同じホーム・ディレクトリを使用して、同じアカウントから Historical Server を起動する必要があります。この制限によって、Historical Server ファイルが不正アクセスによって読み込まれたり、変更されたりするのを防ぎます。

起動コマンドで別のホーム・ディレクトリを指定すると、別のアカウントから Historical Server を起動できます。新しいロケーションに制御ファイルがない場合、Historical Server によって作成されます。ホーム・ディレクトリ内のレコーディング・セッション・データ・ファイルは、別のホーム・ディレクトリを使用する Historical Server では表示できません。

- アカウントに、起動コマンドで指定した Historical Server への検索(実行)アクセスと書き込みアクセスが付与されていること。
- アカウントには、レコーディング・セッション・データ・ファイルのロケーションへの検索(実行)アクセスと書き込みアクセスが必要であること。そしてまた、レコーディング・セッションを作成するユーザがそのアカウントを指定しなければならないこと。デフォルト・ロケーションは Historical Server のホーム・ディレクトリですが、ユーザはレコーディング・セッションの作成時にセッションごとにデフォルトを上書きできます。

## Historical Server のスーパーユーザ・アカウント

Historical Server 起動コマンドの -U パラメータを使用して、オプションで Historical Server のスーパーユーザを指定できます。このパラメータを指定すると、スーパーユーザは次のことを実行できます。

- Historical Server のシャットダウン
- 履歴データ・ファイルの表示または削除

起動コマンドで -U パラメータを指定しない場合、すべてのユーザが Historical Server を停止できますが、履歴データ・ファイルへのアクセスは制限されます。

## Sybase Open Client/Server 接続

Historical Server は Sybase Open Client/Server バージョン 11.1.x でサポートされているため、*interfaces* ファイル(UNIX の場合は *interfaces* ファイル、Windows の場合は *sql.ini* ファイル)またはディレクトリ・サービスを使用してクライアント/サーバ接続を確立します。

ディレクトリ・サービスを使用すると、すべてのクライアント・マシンで *interfaces* ファイルまたは *sql.ini* ファイルを更新する必要がないという利点があります。これらのファイルは、1つのディレクトリ・サービス・エントリによって置き換えられます。サーバのアドレス変更やサーバ名の変更などを簡単に管理できます。

詳細については、『Open Client/Server 設定ガイド UNIX 版』または『Open Client/Server 設定ガイド Windows 版』を参照してください。

## UNIX プラットフォームでの初期設定

この項では、UNIX プラットフォームで Historical Server を設定する方法について説明します。

### UNIX プラットフォームの前提条件

この項で説明する手順は、次の作業が完了していることを想定しています。

- Historical Server ソフトウェアが、配布メディアに記載された手順に従って配布メディアからアンロードされている。
- ネットワーク上で、Adaptive Server と Monitor Server のペアが設定されている。

### UNIX プラットフォームでの設定手順

UNIX プラットフォームで Historical Server を設定するには、次の手順に従います。

- \$SYBASE 環境変数に、Historical Server をアンロードした \$SYBASE インストール・ディレクトリの値を設定します。
- “sybase” アカウント、または \$SYBASE ディレクトリに対する読み込み、書き込み、検索(実行)の各パーミッションが付与されている別のアカウントを使用してログオンします。
- \$PATH 環境変数を設定します。

Historical Server の実行プログラムは \$SYBASE/bin ディレクトリ内にあります。Historical Server を起動するアカウントの \$PATH 環境変数にこのパス名を追加します。

- Historical Server 起動用のスクリプト・ファイルを作成します。  
スクリプト・ファイルによって、各 Historical Server の起動用の正しいパラメータが使用されます。スクリプト・ファイルには、Historical Server 起動コマンド *histserver* とそのパラメータを指定します。

Historical Server スクリプト・ファイルを作成するには、次の手順に従います。

- a エディタを使用して、新しいファイルを作成します。新しいファイルには、次の名前とロケーションを使用することをおすすめします。

*install\_dir/install/run\_histServerName,*

*histServerName* には、Historical Server の名前を指定します。

- b 新しいファイルを編集して、*histserver* コマンドを挿入し、インストール環境に適したパラメータと値を指定します。コマンドの記述には復帰文字を使用しないでください。コマンドが複数行にわたる場合は、UNIX の継続文字 (¥) を使用します。パラメータと値の間のスペースはオプションです。

表 2-1 (21 ページ) は、コマンドとそのパラメータを示します。表内では「必須」とマークされたパラメータは、スクリプト・ファイルで指定する必要があります。デフォルト値を使用する場合は、デフォルト値のついたパラメータを省略できます。

次は、Historical Server を起動するためのスクリプト・ファイルの例です。

```
histserver -Dserver1HistDir -Sserver1Hist ¥
-Usa -Psapasswd ¥
-lserver1HistLog -n15 &
```

- c *chmod* コマンドを使用して、Historical Server を起動するアカウントにスクリプト・ファイルの実行パーミッションを付与します。

- 5 Historical Server の接続情報を追加します。

この作業では、Historical Server にポートまたはネットワーク・アドレスを割り当てます。また、Historical Server が 1 つ以上の Adaptive Server と Monitor Server のペアに確実に接続できるように設定します。

*interfaces* ファイルまたはディレクトリ・サービスに Hisotrical Server 接続情報を追加します。*dsedit*、*dscp*、*dsedit\_dce*、*dscp\_dce* の詳細については、『Open Client/Server 設定ガイド UNIX 版』を参照してください。

*interfaces* ファイルを使用してクライアント／サーバ接続を行う場合、次の手順に従います。

- a Historical Server が使用する *interfaces* ファイルのサーバ・リストをチェックします。Historical Server を実行するには、このファイルに次のすべてのサーバのエントリが含まれている必要があります。

- モニタする Adaptive Server
- 各 Adaptive Server とペアに設定されている Monitor Server
- Historical Server

- b **dsedit** (システムが X-Window を実行している場合) または **dscp** (コマンド・ライン・ユーティリティ) を使用して、*interfaces* ファイルにエントリを追加します。『Open Client/Server 設定ガイド UNIX 版』の手順に従ってください。これらのエントリを追加するには、次の情報が必要です。
  - Historical Server を接続する Monitor Server と Adaptive Server の名前
  - これらのサーバの設定時にサーバに割り当てられたポート番号 またはネットワーク・アドレス。この情報を入手するには、サーバを設定したマシンで **dsedit** または **dscp** を実行し、該当する *interfaces* ファイルを調べる。

ディレクトリ・サービスを使用してクライアント／サーバ接続を行う場合、次の手順に従います。

- Historical Server をインストールしたマシンの *libtcl.cfg* ファイルが適切なディレクトリ・サービスを指していることを確認します。エディタを使用して、*libtcl.cfg* ファイルをチェックし、更新します。
- **dsedit\_dce** (システムが X-Windows を実行している場合) または **dscp\_dce** (コマンド・ライン・ユーティリティ) を使用して、Historical Server をディレクトリ・サービスに追加します。この手順を完了するには、Historical Server 名が必要です。

#### 6 クライアント・マシンで Historical Server を設定します。

この作業によって、クライアントは Historical Server に接続できるようになります。Historical Server クライアントは、レコーディング・セッション または プレイバック・セッションを作成するユーザです。各クライアント・マシンを適切に設定する必要があります。

**dsedit** と **dscp** の使用方法の詳細については、『Open Client/Server 設定ガイド UNIX 版』を参照してください。

*interfaces* ファイルを使用してクライアント／サーバ接続を行う場合、次の手順に従います。

- a Historical Server クライアントが使用するすべての *interfaces* ファイルを更新します。クライアントの *interfaces* ファイルには、次のエントリが必要です。
  - Historical Server
  - Historical Server からモニタする任意の Adaptive Server
  - リストされている各 Adaptive Server に関連付けられた Monitor Server

b dsedit (システムが X-Window を実行している場合) または dscp (コマンド・ライン・ユーティリティ) を使用して、*interfaces* ファイルにエントリを追加します。これらのエントリを追加するには、次の情報が必要です。

- Historical Server 名
- Historical Server を接続する Monitor Server と Adaptive Server の名前
- これらのサーバの設定時にサーバに割り当てられたポート番号 またはネットワーク・アドレス。この情報を入手するには、サーバを設定したマシンで *dsedit* または *dscp* を実行し、該当する *interfaces* ファイルを調べる。

ディレクトリ・サービスを使用してクライアント／サーバ接続を行う場合は、すべての Historical Server クライアント・マシンの *libtcl.cfg* ファイルが適切なディレクトリ・サービスを指していることを確認してください。テキスト・エディタを使用して、*libtcl.cfg* ファイルをチェックし、更新します。

## Windows プラットフォームでの初期設定

この項では、Windows マシンで Historical Server を設定する方法について説明します。この項は、次の項目で構成されています。

- [Windows での前提条件](#)
- [Windows でのインストールの結果](#)
- [Windows での設定手順](#)

### Windows での前提条件

この項の手順では、次の作業が完了していることを想定しています。

- Historical Server ソフトウェアが、配布メディアに記載された手順に従って配布メディアからロードされている。
- ネットワーク上で、Adaptive Server と Monitor Server のペアが設定されている。

## Windows でのインストールの結果

Windows プラットフォームでは、Sybase インストール・プロセスによって、Historical Server のインストールがほとんど完了します。インストール・プロセスは次のことを実行します。

- Historical Server ファイルを Sybase インストール・ディレクトリへコピーする。
- Historical Server 起動コマンドのパラメータ値をレジストリに追加する。これらのパラメータにはデフォルト値が使用される。エントリは次の場所に保存される。

```
HKEY_LOCAL_MACHINE\SOFTWARE\SYBASE\SERVER\  
servername\Parameters
```

- 新しい Historical Server をレジストリのサービスのリストに追加する。エントリは次の場所に保存される。

```
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\  
Services
```

- Historical Server マシン上の Sybase インストール・ディレクトリにある *sql.ini* ファイルに、Historical Server 接続情報を追加する。

## Windows での設定手順

Historical Server の設定を完了するには、次の手順に従います。

- Historical Server の接続情報を追加します。

この作業では、Historical Server にポートまたはネットワーク・アドレスを割り当てます。また、Historical Server が 1 つ以上の Adaptive Server と Monitor Server のペアに確実に接続できるように設定します。*sql.ini* ファイルまたはディレクトリ・サービスに接続情報を追加します。次に説明する手順および *ocscfg* または *dsedit* の詳細については、『Open Client/Server 設定ガイド Windows 版』を参照してください。

*sql.ini* ファイルを使用してクライアント／サーバ接続を行う場合は、*sql.ini* ファイルのサーバ・リストをチェックします。Historical Server を実行するには、このファイルに次のすべてのサーバのエントリが含まれている必要があります。

- モニタする Adaptive Server
- 各 Adaptive Server とペアに設定されている Monitor Server

- Historical Server。Sybase インストール・プロセスによって Historical Server のエントリが追加されるため、Historical Server のエントリはすでに存在している。ただし、Adaptive Server と Monitor Server のペアは通常 Historical Server とは異なるマシン上で実行されるので、Historical Server マシンにはこれらのエントリが存在しない可能性がある。このような場合は、Historical Server が接続する各 Adaptive Server と Monitor Server のペアに対するエントリを追加する必要がある。

`dsedit` を使用して、*sql.ini* ファイルにエントリを追加します。これらのエントリを追加するには、次の情報が必要です。

- Historical Server を接続する Monitor Server と Adaptive Server の名前
- これらのサーバの設定時にサーバに割り当てられたポート番号またはネットワーク・アドレス。この情報を入手するには、サーバを設定したマシンで `dsedit` を実行し、該当する *sql.ini* ファイルまたは *interfaces* ファイルを調べる。

**注意** Adaptive Server 名はエイリアス名にはできません。Monitor Server が認識できる名前を指定する必要があります。たとえば、Monitor Server 起動コマンドの `-S` パラメータで指定した値を使用します。

`dsedit` の代わりにテキスト・エディタを使用して *sql.ini* ファイルを編集する場合は、ファイルの最終行の行末に復帰文字があることを確認してください。

ディレクトリ・サービスを使用してクライアント／サーバ接続を行う場合は、次の手順に従います。

- Historical Server をインストールしたマシンの *libtcl.cfg* ファイルが適切なディレクトリ・サービスを指していることを確認します。`ocscfg` を使用して、*libtcl.cfg* ファイルをチェックし、更新します。
- `dsedit` を使用して、Historical Server を適切なディレクトリ・サービスに追加します。この手順を完了するには、Historical Server 名が必要です。インストール・プロセスで作成されるデフォルト名のフォーマットは、*machineName\_hs* です。たとえば、*smith\_hs* となります。

## 2 Historical Server クライアント・マシンを設定します。

この作業によって、クライアントは Historical Server に接続できるようになります。Historical Server クライアントは、レコーディング・セッションまたはプレイバック・セッションを作成するユーザです。各クライアント・マシンを適切に設定する必要があります。

*sql.ini* ファイルを使用してクライアント／サーバ接続を行う場合は、Historical Server クライアントが使用するすべての *sql.ini* ファイルを更新します。クライアントの *sql.ini* ファイルには、次のエントリが必要です。

- Historical Server
- リストされている各 Adaptive Server に関する付けられた Monitor Server
- Historical Server を使用してデータを収集する Adaptive Server

**dsedit** を使用して、*sql.ini* ファイルにエントリを追加します。これらのエントリを追加するには、次の情報が必要です。

- Historical Server 名
- Historical Server を接続する Monitor Server と Adaptive Server の名前
- これらのサーバの設定時にサーバに割り当てられたポート番号またはネットワーク・アドレス。この情報を入手するには、サーバを設定したマシンで **dsedit** を実行し、該当する *sql.ini* ファイルを調べます。

ディレクトリ・サービスを使用してクライアント／サーバ接続を行う場合は、すべての Historical Server クライアント・マシンの *libtcl.cfg* ファイルが適切なディレクトリ・サービスを指していることを確認してください。

**ocscfg** を使用して、*libtcl.cfg* ファイルをチェックし、更新します。

### 3 レジストリの起動パラメータを確認します。

この作業では、インストール・プロセスによってレジストリに挿入されたデフォルトの起動パラメータ値がサイトに適していることを確認します。

[コントロール パネル] の [サービス] ウィンドウを使用して Historical Server を起動すると、サーバはレジストリ・エントリから起動パラメータを読み込みます。コマンド・ラインまたはバッチ・ファイルからサーバを起動すると、Historical Server はレジストリ・エントリとコマンドの両方の起動パラメータを使用します。両方に同一のパラメータが指定されている場合は、コマンドに指定された値が、レジストリ・エントリの値に優先します。コマンドに起動パラメータを指定していない場合には、デフォルトでレジストリ・エントリのすべてのパラメータが使用されます。

## 起動パラメータの変更

レジストリの起動パラメータやサーバ名を変更するには、次の手順に従います。

- 1 レジストリ・エディタ (*regedit.exe* がこれに該当し、通常は windows ディレクトリの下にある) を起動します。
- 2 Windows の場合は、次のウィンドウを選択します。

¥¥HKEY\_LOCAL\_MACHINE

次に、正しいユーザ・エントリを選択します。

- 3 既存の設定を保存してから、次に進みます。[ファイル]-[エクスポート]を選択し、「選択された部分(E)」に「HKEY\_LOCAL\_MACHINE」が選ばれている事を確認してから、名前を付けて保存します。

- 4 ツリー・ビューで、次のエントリを強調表示します。

```
¥SOFTWARE¥SYBASE¥Server¥
  srvrName¥Parameters
```

*srvrName* は、起動パラメータを変更するサーバの名前です。ウィンドウの右側に表示される、この形式の既存の起動パラメータ・リストを確認します。

```
Argx, dataType, parameter
```

次のことに注意してください。

- *x* は、連続した整数です。
- *dataType* は、パラメータ値のデータ型を定義します。
- *parameter* は起動オプションです。ダッシュで始め、その後にパラメータ値を指定します。

次に、Historical Server の -D、-U、-P 起動パラメータの例を示します。

```
Arg0:REG_SZ:-Dc:¥sybase¥data¥hs_data
Arg1:REG_SZ:-Uhssuper
Arg2:REG_SZ:-Pxdwdr
```

- 5 新しい起動パラメータを追加するには、次の手順に従います。

- [新規] - [文字列値] を選択します。
- 表示されたダイアログ・ボックスの [名前] ボックスに、*Argx* を入力します。*x* は、現在割り当てられていない次の整数です。
- 表示された [データ] ダイアログ・ボックスに、パラメータと値を入力します。

- 6 既存のパラメータを修正するには、次の手順に従います。

- 変更するパラメータ行をダブルクリックします。
- 表示された [値のデータ] ダイアログ・ボックスで、エントリを変更します。

- 7 [ファイル]-[レジストリ エディタの終了] を選択します。

## Historical Server 起動パラメータの設定

この項では、Historical Server の起動コマンドとパラメータについて説明します。この項の内容は、UNIX プラットフォームと Windows プラットフォームで動作する Historical Server に適用されます。

### 関数

Historical Server を起動します。

### 構文

```
histserver -U<user name> -P<password> -D<output dir> -l<log file> -I<interfaces file> [-d<delimiter>] [-O<ASE name>] [-o<DATABASE name>] [-f] -u<output ASE user name> -p<outputASEpassword> [-m]
```

パラメータの意味は次のとおりです。

- -O は、Historical Server のモニタリング・データの出力に使用する Adaptive Server の interfaces ファイル内での名前を指定する。
- -o は、モニタリング・データを格納するデータベースの名前を指定する。
- -u は、出力 Adaptive Server への接続に使用するログイン名を入力する。
- -p は、出力 Adaptive Server に対応するパスワードを入力する。
- -f は、出力ファイルを作成することを指定する (出力 Adaptive Server をすでに指定している場合にも必要)。
- -m は、並列出力モードを使用することを示す。

実行プログラム名は、UNIX プラットフォームの場合は **histserver**、Windows の場合は **histsrvr** です。

### パラメータ

表 2-1 は、Historical Server の起動コマンドのパラメータを示します。

表 2-1: Historical Server 起動パラメータ

パラメータ	説明
<code>-DhomeDir</code>	<p>必須。このパラメータには、Historical Server インスタンスのホーム・ディレクトリを指定します。Historical Server を起動するアカウントには、Historical Server ホーム・ディレクトリでの読み込み、書き込み、検索(実行) パーミッションが必要です。</p> <p>デフォルトのホーム・ディレクトリは、Historical Server を起動するアカウントの現在の作業ディレクトリです。</p> <p>Historical Server のこのインスタンスで使用される制御ファイルは、ホーム・ディレクトリに置かれます。制御ファイルには過去のレコーディング・セッションに関する情報が収められ、現在のセッションの実行中に新しいレコーディング・セッションが確立すると制御ファイルが更新されます。</p> <p>このパラメータで指定するディレクトリは、レコーディング・セッション中に履歴データ・ファイルが書き込まれるデフォルト・ディレクトリになります。レコーディング・セッションを作成するユーザは、制御ファイルの定義に影響を与えることなく、このデフォルト・ロケーションを上書きできます。</p> <p>Historical Server の複数のインスタンスで、同じホーム・ディレクトリを共有できます。詳細については、「<a href="#">Historical Server の複数インスタンスの設定</a>」(23 ページ) を参照してください。</p>
	<p><b>注意</b> ディレクトリ名の最初に UNIX のシェル固有の文字(C シェルの波形記号(~)など)がある場合は、-D パラメータとディレクトリ名を1つ以上のスペース文字で区切ります。スペースで区切らないと、シェルはファイル名の拡張が必要であることを認識できません。</p>
<code>-d delim</code>	<p>Historical Server が生成するビュー・ファイルのデフォルト・ファイルデリミタを指定します。デリミタ(delim)には次の値を指定できます。</p> <ul style="list-style-type: none"> <li>文字。例：<code>./histserver -d ''</code> または <code>./histserver -d/</code></li> <li>エスケープ文字。例：<code>./histserver -d'\'</code></li> <li>デリミタの ASCII コードを表す4桁の16進数。例：<code>./histserver -d0x09</code></li> </ul>
<code>-iinterfacesFile</code> (ダッシュに続く“i” は、大文字と小文字 のどちらも有効)	<p>使用する <i>interfaces</i> ファイルのパス名。ファイルには、Historical Server とこの Historical Server が接続するすべての Adaptive Server と Monitor Server のペアの正しい名前などの接続情報が含まれている必要があります。このパラメータを省略すると、デフォルトの <i>interfaces</i> ファイルの名前とロケーションは次のように設定されます。</p> <p>UNIX : <code>\$SYBASE/interfaces</code></p> <p>Windows の場合 : <code>SYBASE\sql\ini</code> (SYBASE は SYBASE 環境変数の値)</p>
<code>-llogFile</code>	Historical Server に関する情報とエラー・メッセージを記録するログ・ファイルのパス名。起動情報、エラー・メッセージ、トリガされたアラームのレコードなどのデータが収集されます。デフォルトのパス名は、現在のディレクトリの <code>hs.log</code> です。

パラメータ	説明
<b>-UserName</b>	<p>Historical Server スーパユーザの名前。スーパユーザとして Historical Server にログインしたユーザは、レコーディング・セッションを開始したユーザに関係なく、<code>shutdown</code> コマンドを発行して Historical Server を終了したり、Historical Server によって作成されたデータ・ファイルを表示または削除したりできます。</p> <p>このユーザ名は、Adaptive Server のログイン・アカウントまたはオペレーティング・システムの登録名と同じにする必要はありません。詳細については、<a href="#">「Historical Server のスーパユーザ・アカウント」(11 ページ)</a> を参照してください。</p> <p>このパラメータを指定しないと、すべてのユーザが Historical Server を停止できますが、Historical Server のデータ・ファイルへのアクセスは制限されます。</p>
<b>-Ppassword</b>	<p><b>-U</b> パラメータで指定したユーザのパスワード。Historical Server にログインするユーザが Historical Server のスーパユーザ権限を使用するには、この <i>username</i> と <i>password</i> を指定する必要があります。詳細については、<a href="#">「Historical Server のスーパユーザ・アカウント」(11 ページ)</a> を参照してください。</p> <p><b>注意</b> UNIX プラットフォーム上で、このパラメータを指定せずに <b>-U</b> パラメータを指定すると、起動処理中にパスワードの入力を要求するプロンプトが表示されます。ただし、プロンプトを表示するには、Historical Server をフォアグラウンド・モードで起動する必要があります。パスワード入力を要求するプロンプトを表示するには、起動スクリプトを次のように変更します。</p> <ul style="list-style-type: none"> <li>• <b>-P</b> パラメータとアンパサンド (&amp;) を起動スクリプト・ファイルから削除する。</li> <li>• スクリプト・ファイルを実行します。</li> <li>• パスワード入力を要求するプロンプトが表示されたら、<b>-U</b> パラメータで指定した <i>username</i> のパスワードを入力します。</li> <li>• Historical Server プロセスをバックグラウンドで実行します。</li> </ul>
<b>-nmaxConnections</b>	Historical Server から同時に接続できる Open Client の最大数を指定します。有効な値は 1 ~ 20 です。デフォルトは 20 です。
<b>-ShsName</b>	interfaces ファイルで指定されている Historical Server の名前。省略すると、DSLISTEN 環境変数の値が Historical Server のデフォルト名として使用されます。DSLISTEN を設定していない場合は、“histserver”が使用されます。
<b>-v</b>	他のすべてのパラメータを無視して、Historical Server のバージョン情報を表示し、Historical Server を終了します。

## Historical Server の複数インスタンスの設定

Historical Server の複数のインスタンスを実行する場合があります。このような場合は、複数のインスタンスを正しく設定して、複数インスタンスのレコーディング・セッションへのユーザ・アクセスを許可または制限する必要があります。

この項では、次の点について説明します。

- [Historical Server の複数インスタンスの作成時期](#)
- [UNIX プラットフォームでの追加 Historical Server の設定](#)
- [Windows プラットフォームでの追加 Historical Server の設定](#)

### Historical Server の複数インスタンスの作成時期

Historical Server の 1 つのインスタンスを使用して、サンプル・インターバルの短い多数のレコーディング・セッションを同時に管理すると、サンプルの処理と記録に時間がかかり、新しいサンプルをタイムリーに取得できなくなることがあります。記録したデータ・ファイルのタイムスタンプが、指定したサンプル・インターバルでサンプルが収集されていないことを示している場合は、Historical Server の複数インスタンスに負荷を分散できます。

Historical Server の複数のインスタンスが同じホーム・ディレクトリを使用する場合は、次のようになります。

- 各インスタンスは、同じオペレーティング・システムのアカウントを使用して起動する必要がある。
- 同じホーム・ディレクトリを使用するインスタンスはすべて、同じ制御ファイルを使用する。
- 1 つのインスタンスで記録されたセッションは、他のインスタンスにも表示される。
- ファイル・ロック・メカニズムにより、同時アクセスによって制御ファイルが壊れることはない。

Historical Server の複数のインスタンスが異なるホーム・ディレクトリを使用する場合は、次のようになります。

- Historical Server の各インスタンスは、異なるユーザ・アカウントを使用して起動できる。
- Historical Server の各インスタンスは、異なる制御ファイルを使用する。
- 1 つのサーバのレコーディング・セッションは他のサーバには表示されない。たとえば、`isql` から `hs_list` コマンドを発行しても、異なるホーム・ディレクトリを使用する他の Historical Server によって定義されたレコーディング・セッションにはアクセスできない。

複数インスタンスで同じホーム・ディレクトリを使用するか、異なるホーム・ディレクトリを使用するかを決定するには、次の点を検討してください。

- データの可視性 – 複数インスタンスのユーザが他のユーザのレコーディング・セッション定義にアクセスする必要がある場合は、同じホーム・ディレクトリを使用する。共有が望ましくない場合は、異なるホーム・ディレクトリを使用する。プレイバック・セッションへのすべての入力セッションは、同じホーム・ディレクトリにある必要がある。
- ステータス情報の可視性 – `hs_status` コマンドは、同じホーム・ディレクトリを使用する Historical Server の複数のインスタンスを区別できない。Historical Server の管理上、この区別が重要な場合は、異なるホーム・ディレクトリを使用する。
- 制御ファイルの競合 – 大きな制御ファイルに複数ユーザからのアクセスが集中するような場合は、異なるホーム・ディレクトリを使用すると、ユーザへの応答時間が改善されることがある。レコーディング・セッションの作成やレコーディング・セッションに関する情報の一覧表示などのアクションでは、制御ファイルを使用する。レコーディング・セッションの実行中は、レコーディング・セッション自体によって他のユーザが制御ファイルを使用できないようにすることはできない。

## UNIX プラットフォームでの追加 Historical Server の設定

UNIX プラットフォームで追加 Historical Server を設定するには、次の手順に従います。

- オリジナルの Historical Server 用に作成した起動スクリプトをコピーします。新しいファイルに、起動する新しい Historical Server 名を表す名前を付けます。
- 起動ファイルを編集し、Historical Server の新しいインスタンスに合わせて、起動コマンドのパラメータを変更します。詳細については、「[Historical Server 起動パラメータの設定](#)」(20 ページ) を参照してください。Historical Server のホーム・ディレクトリを指定する `-D` パラメータには特に注意してください。Historical Server の複数インスタンスでホーム・ディレクトリを共有するか、異なるホーム・ディレクトリを使用するかを決める必要があります。
- Historical Server の接続情報を設定し、新しい Historical Server のエントリを追加します。新しい Historical Server に、そのマシンに一意のポート番号を使用します。
- クライアント・マシンの接続情報を編集し、新しい Historical Server のエントリを追加します。

## Windows プラットフォームでの追加 Historical Server の設定

サーバ設定ユーティリティでは、新しい Historical Server は設定できません。初期設定後に追加 Historical Server を設定するには、次の手順に従います。

- 1 Windows レジストリへの起動パラメータの追加
- 2 Windows レジストリ・サービス・リストの更新
- 3 Historical Server の接続情報の追加
- 4 Historical Server クライアント・マシンの設定
- 5 .bat ファイルの作成(オプション)

### Windows レジストリへの起動パラメータの追加

レジストリに Historical Server の起動パラメータを追加するには、次の手順に従います。

- 1 レジストリ・エディタ (regedit.exe がこれに該当し、通常は windows ディレクトリの下にある) を起動します。
- 2 Windows 4.0 の場合は、次のウィンドウを選択します。

HKEY\_LOCAL\_MACHINE

次に、適切なユーザを選択します。

- 3 既存の設定を保存してから、次に進みます。[ファイル]-[エクスポート]を選択し、「選択された部分 (E)」に「HKEY\_LOCAL\_MACHINE」が選ばれている事を確認してから、名前を付けて保存します。
- 4 ツリー・ビューで、次のエントリを強調表示します。

¥SOFTWARE¥SYBASE¥Server

- 5 [編集]-[キーの追加] を選択します。
- 6 表示されたダイアログ・ボックスの [キーネーム] ボックスに、追加する Historical Server の名前を入力します。  
[クラス] ボックスは空白のままにしておきます。
- 7 ツリー・ビューで、次のエントリを強調表示します。

¥SOFTWARE¥SYBASE¥Server\\$hsName

hsName は新しい Historical Server の名前です。

- 8 [新規]-[文字列値] を選択します。

- 9 表示されたダイアログ・ボックスの [名前] ボックスに、次のように入力します。

ServerType

表示された [データ] ダイアログ・ボックスに、次のように入力します。

HISServer

- 10 新しい Historical Server 名のエントリが強調表示されていることを確認します。

- 11 [新規] - [キー] を選択します。

- 12 [新しいキー #1] の名前を、次のように変更します。

Parameters

- 13 ツリー・ビューで、次のエントリを強調表示します。

¥SOFTWARE¥SYBASE¥Server¥*hsName*¥Parameters

*hsName* は新しい Historical Server の名前です。

- 14 [新規] - [文字列値] を選択します。

- 15 [値の名前] ボックスに、次のように入力します。

Arg0

REG\_SZ

表示された [データ] ダイアログ・ボックスに、次のように入力します。

-*ShsName*

*hsName* は Historical Server の名前です。

- 16 Historical Server のすべての起動パラメータを入力するまで、手順 13 から 15 までを繰り返します。

少なくとも、次の必須パラメータを入力します。順序は関係ありません。たとえば、-D は Arg1 でなくてもかまいません。

値の名前	データ型	文字列
Arg0	REG_SZ	<p>-ShsName</p> <p><i>hsName</i> には、手順 5 で使用した Historical Server 名を指定する。</p> <p>例：</p> <p style="padding-left: 2em;">-SHS_SERVER1</p>
Arg1	REG_SZ	<p>-DdataDirectoryName</p> <p><i>dataDirectoryName</i> には、Historical Server のデータを格納する既存のディレクトリのフル・パス名を指定する。Historical Server レコーディング・セッションを作成するすべてのユーザには、このディレクトリへの書き込みアクセス権が必要。</p> <p>例：</p> <p style="padding-left: 2em;">-Dc:¥SYBASE¥data¥hs.data</p>
Arg2	REG_SZ	<p>-linstallationRootDir¥ini¥sql.ini</p> <p>例：</p> <p style="padding-left: 2em;">-Ic:¥SYBASE¥ini¥sql.ini</p>
Arg3	REG_SZ	<p>-linstallationRootDir¥install¥logName</p> <p><i>logName</i> には、Historical Server のエラー・メッセージを格納するファイル名を指定する。このファイルがない場合は、起動時にサーバによって作成される。</p> <p>例：</p> <p style="padding-left: 2em;">-lc:¥SYBASE¥install¥hs.log</p>

## Windows レジストリ・サービス・リストの更新

Historical Server をレジストリのサービス・リストに追加するには、次の手順に従います。

- 1 レジストリ・エディタ (*regedit.exe* がこれに該当し、通常は windows ディレクトリの下にある) を起動します。
- 2 Windows 4.0 の場合は、次のウィンドウを選択します。  
HKEY\_LOCAL\_MACHINE  
次に、正しいユーザを選択します。
- 3 既存の設定を保存してから、次に進みます。[ファイル]-[エクスポート]を選択し、「選択された部分 (E)」に「HKEY\_LOCAL\_MACHINE」が選ばれている事を確認してから、名前を付けて保存します。
- 4 ツリー・ビューで、次のエントリを強調表示します。

¥SYSTEM¥CurrentControlSet¥Services

- 5 [新規] - [キー] の名前を、次のように変更します。
- 6 [新しいキー #1] ボックスに、次のように入力します。

SYBHIS\_ *hsName*

*hsName* には、直前に設定した Historical Server の名前を指定します。

- 7 ツリー・ビューで、次のエントリを強調表示します。

¥SYSTEM¥CurrentControlSet¥Services¥SYBHIS\_ *hsName*

- 8 SYBHIS\_ *hsName* キー・エントリに、次の値、データ型、文字列をすべて入力するまで、[編集] - [値の追加] を繰り返し選択します。

値の名前	データ型	文字列
DisplayName	REG_SZ	Sybase HIServer_ <i>hsName</i>
ErrorControl	REG_DWORD	01 (16 進ラジオ・ボタンを選択)
Group	REG_SZ	(空白)
ImagePath	REG_EXPAND_SZ	rootInstlDir¥bin¥histsrvr.exe -ShsName -C
ObjectName	REG_SZ	LocalSystem
Start	REG_DWORD	03 (16 進ラジオ・ボタンを選択)
Type	REG_DWORD	010 (16 進ラジオ・ボタンを選択)

- 9 SYBHIS\_ *hsName* キー・エントリが強調表示されていることを確認します。
- 10 [編集] - [キーの追加] を選択します。
- 11 [新しいキー #1] の名前を、次のように変更します。

Security

[クラス] ボックスは空白のままにしておきます。

- 12 ツリー・ビューで、次の Security キー・エントリを強調表示します。

¥SOFTWARE¥SYBASE¥Server¥*hsName*¥Security

- 13 [新規] - [バイナリ値] を選択します。
- 14 値の名前、データ型、バイナリ・エディタに次の値を入力します。

値の名前	データ型	バイナリ・エディタ
Security	REG_BINARY	01001480

- 15 [ファイル] - [レジストリ エディタの終了] を選択します。
- 16 マシンを再起動します。

## Historical Server の接続情報の追加

この作業では、Historical Server にポートまたはネットワーク・アドレスを割り当てます。また、Historical Server が 1 つ以上の Adaptive Server と Monitor Server のペアに確実に接続できるように設定します。

この作業は、クライアント／サーバ接続に *sql.ini* ファイルまたはディレクトリ・サービスのどちらを使用するかによって異なります。

- *sql.ini* ファイルを使用してクライアント／サーバ接続を行う場合は、*sql.ini* ファイルで Historical Server マシンのサーバ・リストをチェックします。Historical Server を実行するには、このファイルに次のすべてのサーバのエントリが含まれている必要があります。
  - モニタする Adaptive Server
  - 各 Adaptive Server とペアに設定されている Monitor Server
  - Historical Server

**dsedit** を使用して、*sql.ini* ファイルにエントリを追加します。これらのエントリを追加するには、次の情報が必要です。

- Historical Server を接続する Monitor Server と Adaptive Server の名前
- これらのサーバの設定時にサーバに割り当てられたポート番号またはネットワーク・アドレス。

この情報を入手するには、サーバを設定したマシンで **dsedit** を実行し、該当する *sql.ini* ファイルを調べます。

---

**注意** Adaptive Server 名はエイリアス名にはできません。Monitor Server が認識できる名前を指定する必要があります。たとえば、Monitor Server 起動コマンドの **-S** パラメータで指定した値を使用します。

---

- ディレクトリ・サービスを使用してクライアント／サーバ接続を行う場合、次の手順に従います。
  - Historical Server をインストールしたマシンの *libtcl.cfg* ファイルが適切なディレクトリ・サービスを指していることを確認します。**ocscfg** を使用して、*libtcl.cfg* ファイルをチェックし、更新します。
  - **dsedit** を使用して、Historical Server を適切なディレクトリ・サービスに追加します。この手順を完了するには、Historical Server 名が必要です。インストール・プロセスで作成されるデフォルト名のフォーマットは、*machineName\_hs* です。たとえば、*smith\_hs* となります。

**ocscfg** と **dsedit** の使用方法については、『Open Client/Server 設定ガイド デスクトップ・プラットフォーム版』を参照してください。

## Historical Server クライアント・マシンの設定

この作業によって、クライアントは Historical Server に接続できるようになります。Historical Server クライアントは、レコーディング・セッションまたはプレイバック・セッションを作成するユーザです。各クライアント・マシンを適切に設定する必要があります。

この設定作業は、クライアント/サーバ接続に *sql.ini* ファイルまたはディレクトリ・サービスのどちらを使用するかによって異なります。

- *sql.ini* ファイルを使用してクライアント/サーバ接続を行う場合は、Historical Server クライアントが使用するすべての *sql.ini* ファイルを更新します。クライアントの *sql.ini* ファイルには、次のエントリが必要です。
  - Historical Server
  - リストされている各 に関連付けられた Monitor Server
  - Historical Server を使用してデータを収集する Adaptive Server

**dsedit** を使用して、*sql.ini* ファイルにエントリを追加します。これらのエントリを追加するには、次の情報が必要です。

- Historical Server 名
- Historical Server を接続する Monitor Server と Adaptive Server の名前
- これらのサーバの設定時にサーバに割り当てられたポート番号またはネットワーク・アドレス。

この情報を入手するには、サーバを設定したマシンで **dsedit** を実行し、該当する *sql.ini* ファイルを調べます。

- ディレクトリ・サービスを使用してクライアント/サーバ接続を行う場合は、すべての Historical Server クライアント・マシンの *libtcl.cfg* ファイルが適切なディレクトリ・サービスを指していることを確認してください。  
**ocscfg** を使用して、*libtcl.cfg* ファイルをチェックし、更新します。

### .bat ファイルの作成 (オプション)

この作業では、サービス・アイコンを使用する代わりにバッチ・ファイルを実行して Historical Server を起動できます。

バッチ・ファイルから Historical Server を起動するには、次の手順を実行します。

- 1 テキスト・エディタを起動し、新しいテキスト・ファイルを開きます。
- 2 Historical Server の起動コマンド `histsrvr` と必要なすべてのパラメータを入力します。起動コマンド `histsrvr.exe` のフル・パス名を使用します。Historical Server が起動パラメータの取得に使用する優先度規則については、「[レジストリからの起動パラメータの推定 \(37 ページ\)](#)」を参照してください。起動パラメータの詳細については、「[Historical Server 起動パラメータの設定 \(20 ページ\)](#)」を参照してください。

次に、.bat ファイルの行の例を示します。この例では、コマンドに復帰文字が挿入されています。実際の .bat ファイルには、復帰文字を使用しないでください。ファイル全体を 1 行で表す必要があります。

```
c:\$sybase\$bin\$histsrvr.exe -Shs_server1  
-Dc:\$sybase\$data\$hs_data  
-Ic:\$sybase\$ini\$sql.ini  
-lc:\$sybase\$data\$hs.log
```

- 3 .bat 拡張子を付けてファイルを保存します。RUN\_ *hsName.bat* という名前の使用をおすすめします。次に例を示します。

```
RUN_hs_server1.bat
```



トピック名	ページ
<a href="#">UNIX プラットフォームでの Historical Server の起動と停止</a>	33
<a href="#">Windows での Historical Server の起動と停止</a>	36

## UNIX プラットフォームでの Historical Server の起動と停止

この項では、UNIX プラットフォームで実行する Historical Server の起動と停止の方法について説明します。

### UNIX での Historical Server の起動

UNIX では、次の 2 とおりの方法で設定済みの Historical Server を起動できます。

- UNIX シェル・プロンプトから `histserver` コマンドを実行する方法。この方法を使用する場合は、コマンドを実行するたびに適切なパラメータをすべて入力してください。
- `histserver` コマンドと必要なすべてのパラメータを含むスクリプト・ファイルを実行する方法。「[第 2 章 Historical Server の設定](#)」の設定指示に従った場合は、次のコマンドを使って Historical Server を起動します。

*install\_dir/install/RUN\_histServerName*

*install\_dir* は Sybase のルート・ディレクトリです。*histServerName* は起動する Historical Server の名前です。

どちらの方法を使用する場合にも、次のことを守ってください。

- モニタする Adaptive Server と、それに対応する Monitor Server が実行中であることを確認すること。
- 以前に記録したセッションにアクセスする新しいインスタンスが必要な場合は、Historical Server を起動するたびに同じアカウントを使用すること。詳細については、「[オペレーティング・システムの起動アカウント](#) (10 ページ)」を参照してください。
- \$SYBASE 環境変数を Sybase のインストール先のルート・ディレクトリに設定すること。

\$SYBASE 環境変数には、Historical Server の該当する *locales* と *charsets* の各サブディレクトリを持つディレクトリの名前が含まれている必要があります。これらのサブディレクトリは、インストール時に自動的に作成され、移植されます。

\$SYBASE 環境変数は、*interfaces* ファイルのデフォルト・ロケーションも識別します。このデフォルト・ロケーションを上書きするには、*histserver* コマンドに対してパラメータを使用します。

Historical Server は、起動処理が正常に行われると、次のようなメッセージを表示します。

```
Initialization is over. Ready to accept connections.
```

Historical Server は、起動処理中にメッセージをログ・ファイルに書き込みます。正常に起動した場合は、これらのメッセージを無視して構いません。正常に起動しなかった場合は、このログ・ファイルを検査して問題を調べます。

Historical Server ログ・ファイルのデフォルトのパス名は、起動時のディレクトリの *hs.log* です。このデフォルトのパス名は、*histserver* コマンドの *-l* パラメタ (英字の *l*) を使用して変更できます。

## UNIX での Historical Server の停止

この項では、次の項目について説明します。

- [Historical Server をシャットダウンできるユーザ](#)
- [Historical Server の現在のアクティビティの決定](#)
- [遅延シャットダウンと即時シャットダウン](#)
- [シャットダウン手順の詳細](#)

### Historical Server をシャットダウンできるユーザ

起動コマンドにスーパユーザ・アカウントを指定すると、Historical Server を停止できるのはスーパユーザだけとなります。起動コマンドでスーパユーザを指定しないと、すべてのユーザが Historical Server を停止できます。

スーパユーザは、Historical Server 起動コマンドの *-U* パラメータと *-P* パラメタに指定されたアカウントのユーザです。

### Historical Server の現在のアクティビティの決定

Historical Server をシャットダウンする前に、Historical Server の現在のアクティビティを確認し、遅延シャットダウンと即時シャットダウンのどちらを実行するかを決定してください。遅延シャットダウンでは、現在のアクティビティをすべて完了してから Historical Server を終了します。

Historical Server のアクティビティには、完全に定義済みで起動済みのレコーディング・セッション、定義作業が進行中で起動されていないレコーディング・セッション、進行中のプレイバック・セッション、定義中のプレイバック・セッションがあります。クライアント接続は複数のマシンから実行でき、各種の Adaptive Server をモニタリングします。

Historical Server の現在のアクティビティを決定するには、**isql** ユーティリティを使用して Historical Server に接続し、**hs\_status activity** コマンドを発行します。

## 遅延シャットダウンと即時シャットダウン

Historical Server をクローズするには、**isql** ユーティリティを使用して Historical Server に接続し、次のコマンドのいずれかを発行します。

- **hs\_shutdown** – アクティブなレコーディング・セッションがすべて完了し、その他のアクティブな接続がすべてクローズするまでシャットダウンを遅らせます。この間、新しい接続は許可されません。

遅延シャットダウンは、一定時間コマンドの実行をブロックします。**hs\_shutdown** コマンドがブロックされている間は、[Ctrl] キーを押しながら [C] キーを押しても何も起こりません。

- **hs\_shutdown no\_wait** – Historical Server を即時にシャットダウンし、すべての接続とアクティブなレコーディング・セッションを終了します。

どちらの場合も、アクティブなレコーディング・セッションは、制御された方法でシャットダウンします。Historical Server の制御ファイルと履歴データ・ファイルは、Historical Server を再起動すると使用できるようになります。

## シャットダウン手順の詳細

次の順序でサーバをシャットダウンします。

- Historical Server
- Monitor Server
- Adaptive Server

UNIX プラットフォーム上で Historical Server を停止するには、次の手順に従います。

- 1 Historical Server に接続します。**isql** を使用している場合は、次のコマンドを使用します。

```
isql [ -Uhs_superuser_name -Phs_superuser_password ]
      -Shistorical_server
```

各パラメータの意味は、次のとおりです。

- *superuser\_name* は、Historical Server 起動コマンドの -U パラメータで指定した名前です。起動コマンドで -U を指定していない場合は、すべてのユーザが Historical Server を停止できます。この `isql` コマンドの -U パラメータはオプションです。
  - *superuser\_password* は、Historical Server 起動コマンドの -P パラメータで指定したパスワードです。起動コマンドで -U を指定していない場合は、すべてのユーザが Historical Server を停止できます。このパラメータはオプションです。
  - *historical\_server* は、停止する Historical Server の名前です。
- 2 Historical Server の現在のアクティビティを決定するには、`isql` プロンプトに対して次のコマンドを発行します。

```
1> hs_status activity
2> go
```

- 3 Historical Server をシャットダウンするには、次のいずれかのコマンドを発行します。

```
1> hs_shutdown
2> go
```

または

```
1> hs_shutdown no_wait
2> go
```

## Windows での Historical Server の起動と停止

この項では、Windows プラットフォームで実行する Historical Server の起動と停止の方法について説明します。次の項目について説明します。

- [Windows での Historical Server の起動](#)
- [レジストリからの起動パラメータの推定](#)
- [Historical Server が実行中であることの確認](#)
- [Windows での Historical Server の停止](#)

## Windows での Historical Server の起動

次の方法で Historical Server を起動できます。

- Windows の [コントロール パネル] の [サービス] ウィンドウ。
- 起動コマンドとパラメータが入っているバッチ (.bat) ファイル。ファイル名は *run\_hsName.bat* です。hsName は Historical Server インスタンスの名前です。Sybase では、ファイル・マネージャでバッチ・ファイルをダブルクリックする代わりに、コマンド・ライン・シェルからバッチ・ファイルを呼び出すことをおすすめします。コマンド・ライン・シェルは、起動時にエラーが発生した場合にエラー・メッセージを表示しますが、ファイル・マネージャはエラー・メッセージを表示しません。
- コマンド・ライン・シェルから直接入力された起動コマンド。Windows の Historical Server 実行ファイル名は *histsrvr.exe* です。

.bat ファイルまたはコマンド・ラインのコマンドを使用して Historical Server を起動すると、サーバ・プロセスは現在のログイン・アカウントにリンクされます。ログオフすると、サーバはシャット・ダウンします。

運用システムについては、Sybase では Windows の [コントロール パネル] の [サービス] を使用して Historical Server を起動することをおすすめします。サーバをサービスとして起動すると、ログインしている間はその状態が保持されます。

## レジストリからの起動パラメータの推定

Historical Server の起動パラメータは、次のレジストリ・エントリにあります。

```
¥SOFTWARE¥SYBASE¥SERVERTYservername¥Parameters
```

[コントロール パネル] を使用して Historical Server を起動すると、サーバはこのレジストリ・エントリから起動パラメータを読み込みます。コマンド・ラインまたはバッチ・ファイルを使用してサーバを起動すると、起動パラメータはこのレジストリ・エントリと入力コマンドの両方から読み込まれます。両方に同一のパラメータが指定されている場合は、コマンドに指定された値が、レジストリ・エントリの値に優先します。コマンドに起動パラメータを指定していない場合には、デフォルトでレジストリ・エントリのすべてのパラメータが使用されます。

レジストリ・エントリの編集については、[「Windows レジストリへの起動パラメータの追加」\(25 ページ\)](#) を参照してください。

## Historical Server が実行中であることの確認

Windows では、Windows の [コントロール パネル] の [サービス] ウィンドウで Historical Server の状態を確認します。

## Windows での Historical Server の停止

この項では、次の項目について説明します。

- [Windows で Historical Server をシャットダウンできるユーザ](#)
- [Windows での現在のアクティビティの決定](#)
- [Windows での遅延シャットダウンと即時シャットダウン](#)
- [Windows でのシャットダウン手順の詳細](#)
- [Windows 上で避けるべきシャットダウン方法](#)

### Windows で Historical Server をシャットダウンできるユーザ

起動時にスーパーユーザ・アカウントを指定すると、Historical Server を停止できるのは、スーパーユーザ・アカウントだけとなります。起動コマンドでスーパーユーザを指定しないと、すべてのユーザが Historical Server を停止できます。[コントロール パネル] の [サービス] の使用権限を持つ Administrator も、Historical Server をシャットダウンできます。

スーパーユーザ・アカウントは、Historical Server 起動コマンドの -U パラメータと -P パラメータで指定します。

### Windows での現在のアクティビティの決定

Historical Server をシャットダウンする前に、Historical Server の現在のアクティビティを確認し、遅延シャットダウンと即時シャットダウンのどちらを実行するかを決定してください。遅延シャットダウンでは、現在のアクティビティをすべて完了してから Historical Server を終了します。

Historical Server のアクティビティには、完全に定義済みで起動済みのレコーディング・セッション、定義作業が進行中で起動されていないレコーディング・セッション、進行中のプレイバック・セッション、定義中のプレイバック・セッションがあります。クライアント接続は複数のマシンから実行でき、各種の Adaptive Server をモニタリングします。

Historical Server の現在のアクティビティを決定するには、`isql` ユーティリティを使用して Historical Server に接続し、`hs_status activity` コマンドを発行します。

### Windows での遅延シャットダウンと即時シャットダウン

Historical Server をクローズするには、`isql` ユーティリティを使用して Historical Server に接続し、次のコマンドのいずれかを発行します。

- `hs_shutdown` – アクティブなレコーディング・セッションがすべて完了し、他のアクティブな接続がすべてクローズするまでシャットダウンを遅らせます。この間、新しい接続は許可されません。

遅延シャットダウンは、一定時間コマンドの実行をブロックします。 `hs_shutdown` コマンドが `isql` 内でブロックされている間は、[Ctrl] キーを押しながら [C] キーを押しても何も起りません。

- `hs_shutdown no_wait` – Historical Server を即時にシャットダウンし、すべての接続とアクティブなレコーディング・セッションを終了します。 どちらの場合も、アクティブなレコーディング・セッションは、制御された方法でシャットダウンします。 Historical Server の制御ファイルと履歴データ・ファイルは、Historical Server を再起動すると使用できるようになります。

## Windows でのシャットダウン手順の詳細

次の順序でサーバをシャットダウンします。

- Historical Server
- Monitor Server
- Adaptive Server

Windows 上で Historical Server を停止するには、次のいずれかの方法を使用します。

- Windows の [コントロール パネル] の [サービス] ウィンドウで、[停止] ボタンをクリックする。
- `isql` から `hs_shutdown` コマンドを発行する。
- コマンド・ラインまたはバッチ・ファイルを使用してサーバを起動すると、Historical Server はサーバを起動したログインに関連付けられる。この場合、Historical Server を起動したウィンドウを閉じたり、システムからログオフしたりすると、Historical Server がシャットダウンする。

Historical Server の制御シャットダウンの処理時間が、事前に設定されている時間間隔(ウィンドウを閉じるまでの 5 秒間、またはログオフまでの 20 秒間)よりも長いと、システムは通常の時間間隔で、処理を終了するかどうかを確認するポップアップ・ダイアログ・ボックスを表示します。

---

**警告！** このダイアログ・ボックスが表示されるたびに [待機] を選択しないと、Historical Server は制御されずにシャットダウンするため、データの損失やファイルの破壊につながる可能性があります。

---

`isql` を使用して Historical Server を停止するには、次の手順に従います。

- 1 `isql` を使用して Historical Server に接続します。

```
isql -Uhs_supersuer_name -Phs_superuser_password
      -Shistorical_server
```

各パラメータの意味は、次のとおりです。

- *superuser\_name* は、Historical Server 起動コマンドの -U パラメータで指定した名前です。起動コマンドで -U を指定していない場合は、すべてのユーザが Historical Server を停止できます。このパラメータはオプションです。
  - *superuser\_password* は、Historical Server 起動コマンドの -P パラメータで指定したパスワードです。起動コマンドで -U を指定していない場合は、すべてのユーザが Historical Server を停止できます。このパラメータはオプションです。
  - *historical\_server* は、停止する Historical Server の名前です。
- 2 Historical Server の現在のアクティビティを決定するには、**isql** プロンプトに対して次のコマンドを発行します。

```
1> hs_status activity  
2> go
```

- 3 **isql** プロンプトが表示されたら、次のいずれかのコマンドを発行します。

```
1> hs_shutdown  
2> go
```

または

```
1> hs_shutdown no_wait  
2> go
```

## Windows 上で避けるべきシャットダウン方法

Historical Server を最初に シャットダウンしないでシステムをシャットダウンすると、制御されずにシャットダウンされることがあります。Historical Server がきわめて高いレベルのレコーディング・アクティビティを実行している場合を除き、制御シャットダウンは 20 秒以内に終了します。制御シャットダウンの処理に 20 秒以上かかる場合、システム介入によって Historical Server が制御されない形で終了することがあります。

---

**警告！** 安全を期するため、Sybase では Historical Server を手動で停止してからシステムをシャットダウンするようおすすめします。

[強制終了] ボタンを使用すると、制御されないシャットダウンが発生します。

---

この章では、Historical Server のコマンド・インターフェースについて説明します。

トピック名	ページ
<a href="#">コマンドの概要</a>	41
<a href="#">コマンドの構文</a>	42
<a href="#">コマンド・ステータスとコマンド・エラー</a>	43
<a href="#">Historical Server への入力手段としてのスクリプト・ファイル</a>	44
<a href="#">Historical Server への接続</a>	44
<a href="#">Historical Server のコマンド</a>	48

## コマンドの概要

表 4-1 は、Historical Server のコマンドをまとめたものです。

表 4-1: Historical Server のコマンド

アクティビティ	コマンド
レコーディング・セッションの作成	<p>レコーディング・セッションを作成するには、以下のコマンドを使用する。</p> <ul style="list-style-type: none"> <li>• <code>hs_create_recording_session</code> – レコーディング・セッションの特性を定義する。</li> <li>• <code>hs_create_view</code> – ビュー（記録するデータのコレクション）を定義する。</li> <li>• <code>hs_create_filter</code> – データ項目のフィルタリング基準を指定する。</li> <li>• <code>hs_create_alarm</code> – アラーム・アクションをトリガするデータ項目のスレッショルド値を指定する。</li> <li>• <code>hs_initiate_recording</code> – レコーディング・セッションの定義を終了し、あらかじめ指定した開始時刻にレコーディングを開始できるようにする。</li> <li>• <code>hs_terminate_recording</code> – レコーディング・セッションを停止する。</li> </ul>
レコーディング・セッション定義の確認	• <code>hs_list</code> – レコーディング・セッションに対して定義されたすべてのビューとそれらのデータ項目をリストする。

アクティビティ	コマンド
プレイバック・セッションの作成	<p>プレイバック・セッションを作成するには、以下のコマンドを使用する。</p> <ul style="list-style-type: none"> <li>• <code>hs_create_playback_session</code> – プレイバック・セッションの特性を定義する。</li> <li>• <code>hs_create_playback_view</code> – 入力レコーディング・セッションから 1 つ以上のビューを指定して、プレイバック・セッションに含める。</li> <li>• <code>hs_initiate_playback</code> – プレイバック・セッションの定義を終了する。また、ファイルに対し、プレイバックを開始する。</li> <li>• <code>hs_playback_sample</code> – クライアントに対し、サンプルをプレイバックする。</li> <li>• <code>hs_terminate_playback</code> – プレイバック・セッションを停止する。</li> </ul>
Historical Server の管理	<p>Historical Server の管理を行うには、以下のコマンドを使用する。</p> <ul style="list-style-type: none"> <li>• <code>hs_delete_data</code> – 指定のレコーディング・セッションの記録されたデータを含むファイルを削除する。</li> <li>• <code>hs_status</code> – Historical Server の現在のステータスを表示する。</li> <li>• <code>hs_shutdown</code> – Historical Server を停止する。</li> </ul>

## コマンドの構文

Historical Server のコマンドの構文は次のとおりです。

`hs_XXX arg1, arg2, arg3, ..., argn`

`XXX` にはコマンド名を指定します。たとえば、典型的なコマンドには `hs_create_recording_session` があります。

複数のパラメータを指定する場合、パラメータ間をカンマで区切ります。オプションのパラメータやデフォルト値を持つパラメータは省略できます。ただし、それらのパラメータの後に他のパラメータの値が続く場合は、プレースホルダとしてそれらのパラメータに NULL 値を使用します。たとえば、次のコマンドは、ほとんどのパラメータにデフォルト値を使用します。

```
hs_create_playback_session null, null, raw, null, null,
client, null, null, null, null, 7
```

コマンド名、コマンド・キーワード、単語 `null`、データ項目、統計タイプはすべて、大文字と小文字が区別されます。ファイル名、ビュー名、ユーザが指定するその他の名前は、大文字と小文字が区別されます。

パラメータ値に埋め込みスペース (データ項目、統計タイプ、日時指定における埋め込みスペースなど) が含まれる場合は、その値を引用符で囲みます。正しく閉じられた 1 組の一重引用符または二重引用符は、有効なデリミタになります。

パラメータ値に、値全体を区切るために使用されている文字と同じ埋め込み引用符が含まれる場合、そのパラメータ値の中に 1 組の引用符を使用します。この 1 組の引用符は、Historical Server によって 1 文字に圧縮されます。

引用符の中の `null` という単語は、キーワードではありません。

Historical Server のコマンドは複数行にわたって入力することもできます。

## コマンド・ステータスとコマンド・エラー

すべての Historical Server のコマンドは、0 または 1 のいずれかのステータス値を返します。

- 0 は、正常に実行されたことを示します。
- 1 は、エラー状態であることを示します。ステータス “1” が返された場合、エラー・メッセージとそれに対応するエラー・コード、重大度レベル、状態がクライアントに返されます。

Historical Server は、Client-Library™ を使用して 1 組以上の Adaptive Server と Monitor Server との間で通信する Open Server アプリケーションです。これらのコンポーネントのどちらも、エラー状態が検出、報告されることがあります。

エラー状態は、Historical Server からも検出、報告され、クライアントに対して記録と報告のいずれかまたは両方が実行されます。Historical Server のエラー・コードはすべて、30000 から 30999 までの 5 桁の数字です。Open Server のエラー・コードは、16000 から 20000 までの範囲です。エラー・コードがこれらの範囲のどちらかに含まれる場合、Historical Server は以下の重大度レベルのいずれかを割り当てます。

表 4-2: コマンド・エラーの重大度レベル

重大度	説明
1	情報メッセージ
2	警告メッセージ
3	致命的なサーバ・エラー
4	致命的なプロセス・エラー
5	オペレーティング・システム・エラー

Client-Library、Adaptive Server、Monitor Server により検出されたエラーには、それぞれの製品独自のエラー番号付けスキームと重大度レベル・スキームが使用されます。

## Historical Server への入力手段としてのスクリプト・ファイル

Historical Server に入力する場合、コマンドをテキスト・ファイルに入力し、そのファイルを入力手段として使用すると便利です。Historical Server への入力ファイルは `isql` ファイルではなく、`isql` 規則に従う必要もありません。たとえば、Historical Server にはコメント機能はありません。

「[付録 D レコーディング・セッション・ビューの例](#)」には、入力ファイル作成の参考として使用できるビューの例が記載されています。これらのビューは、インストール・ディレクトリの `sample/histserver` サブディレクトリにインストールされている `views` ファイルにもあります。しかし、このファイルには説明テキストが含まれており、これを削除してからでなければ、Historical Server への入力手段として使用することができません。

## Historical Server への接続

この項では、次の項目について説明します。

- 接続前の前提条件
- 接続方法
- Historical Server のアクティビティに必要なパーミッション
- 相互に排他的なセッション

### 接続前の前提条件

Historical Server に接続する前に、次の条件が満たされていることを確認します。

- パフォーマンス・データの収集対象となる Adaptive Server が実行中であること。
- Adaptive Server をモニタしている Monitor Server が実行中であること。Adaptive Server と Monitor Server が実行中かどうかを確認する手順については、『ASE Monitor Server ユーザーズ・ガイド』を参照してください。
- Historical Server が実行中であること。Historical Server が実行中であるかどうかを確認する手順については、[「第 3 章 Historical Server の起動と停止」](#) を参照してください。
- Historical Server マシンとそのクライアント・マシンについて、接続情報が正確に設定されていること。詳細については、[「第 2 章 Historical Server の設定」](#) を参照してください。

## 接続方法

Historical Server へは、SQL Advantage や `isql` など、Sybase サーバへの対話型インターフェースを提供するユーティリティを使用して接続します。ユーティリティの標準的な接続手順に従ってください。たとえば、SQL Advantage では、[接続] メニュー項目を選択します。`isql` では、-S パラメータの値として Historical Server 名を使用して、`isql` を起動します。

## Historical Server のアクティビティに必要なパーミッション

表 4-3 に示すように、実行したいアクティビティに対するパーミッションを持つユーザ・ログインとパスワードを使用して Historical Server に接続します。必要とされるパーミッションによっては、異なるアクティビティを実行するために、切断および異なるログイン・アカウントでの再接続を行うことが必要になる場合があります。

表 4-3: Historical Server の各アクティビティに必要なパーミッション

アクティビティ	アクティビティの実行に必要なパーミッション
次のスーパーユーザ・アクティビティ。 <ul style="list-style-type: none"><li>Historical Server を停止する。</li><li>レコーディング・セッション・データ・ファイルなどの Historical Server ファイルを削除する。</li><li>プライベート・レコーディング・セッションからファイルにアクセスする。</li></ul>	スーパーユーザ・パーミッション。スーパーユーザ・パーミッションを得るために、起動時に定義されたスーパーユーザ・アカウントを使用して Historical Server に接続する。スーパーユーザ・アカウントは、すべての Adaptive Server において有効なアカウントである必要はない。
レコーディング・セッションを作成する。	モニタしている Adaptive Server において有効なログインとパスワード。
( <code>protection_level</code> パラメータが <b>public</b> であるレコーディング・セッションから) パブリック・レコーディング・セッション・ファイルを使用して、ブレイバッブ・セッションを作成する。 パブリック・レコーディング・セッション・ファイル上で <code>hs_list</code> コマンドを使用する。	モニタしている Adaptive Server の <code>mon_rpc_connect</code> ストアド・プロシージャの実行パーミッション。このパーミッションは、Monitor Server に接続するために必要。
	モニタしている Adaptive Server において有効なログインとパスワード。

アクティビティ	アクティビティの実行に必要なパーミッション
<p>(<i>protection_level</i> パラメータが <b>private</b> であるレコーディング・セッションから) プライベート・レコーディング・セッション・ファイルを使用して、プレイバック・セッションを作成する。</p> <p>プライベート・レコーディング・セッション・ファイル上で <code>hs_list</code> コマンドを使用する。</p>	このプライベート・セッションを作成したものと同じログイン、またはスーパーユーザ。後者の場合、スーパーユーザ・アカウントは、モニタしている Adaptive Server において有効なログイン名でなければならない。

1台の Historical Server で、Adaptive Server と Monitor Server の複数のペアに接続し、データを収集することができます。たとえば、1つの Historical Server 接続から、2つのレコーディング・セッション (server1\_MON という Monitor Server を使用して server1 という Adaptive Server のパフォーマンス・データを収集するレコーディング・セッションと、server2\_MON という Monitor Server を使用して server2 という Adaptive Server のデータを収集するレコーディング・セッション) を作成できます。双方の Adaptive Server インスタンスに有効なログイン・アカウントを持っていない場合は、異なるログイン・アカウントを使用した、Historical Server への異なる接続の中で、これらのレコーディング・セッションを作成する必要があります。ただし、この2つのレコーディング・セッションからのデータ・ファイルは、同じディレクトリに保存できます。

`hs_create_recording_session` コマンドは、接続先の Monitor Server を指定します。これは間接的に、パフォーマンス情報の入手対象となる Adaptive Server を示すことになります。

## 相互に排他的なセッション

Historical Server 接続は、一度に1セッションにしか関与できません。レコーディング・セッションの定義と、プレイバック・セッションの定義と実行のいずれかを行うことはできますが、複数のセッションに対するコマンドを混在させることはできません。つまり、`hs_create_recording_session` コマンドを使用してレコーディング・セッションの定義を開始した場合、コマンドのシーケンスを終了してそのセッションを定義(またはキャンセル)してからでなければ、プレイバック・セッションや新しいレコーディング・セッションの定義を開始することはできません。

レコーディング・セッションの定義は、進行中のレコーディング・セッションと同じものではありません。レコーディング・セッションを定義するユーザは、レコーディング・セッションを定義する間に、Historical Server に接続され、Historical Server との間でアクティブな通信を行います。このユーザは、実際のレコーディングの間は、Historical Server との間でアクティブな通信を行いません。レコーディング時には、どのユーザ接続も関係しません。たとえば、ユーザが午後3時に Historical Server に接続し、午後10時からレコーディング・セッションの定義を開始し、その後 Historical Server との接続を切断したとします。午後10時になったとき、ユーザが Historical Server に接続されているかどうかは関係しません。Historical Server が実行中であればレコーディング・セッションが発生します。

しかし、プレイバック・セッションでは、プレイバックの定義とプレイバックの実行のどちらの間でもユーザ接続が必要になります。プレイバックの実行をキャンセルし、ユーザ接続を使用して他の通信を実行できるようにするには、ユーザが `hs_terminate_playback` コマンドを発行します。

## レコーディング・セッションの定義の開始／終了

`hs_create_recording_session` コマンドは、レコーディング・セッションの定義を開始します。このコマンドを発行してから、レコーディング・セッション用のビュー、フィルタ、アラームを作成する他のコマンドを発行することで、レコーディング・セッションの詳細を定義します。

レコーディング・セッションの定義を終了するには、次のいずれかのコマンドを使用します。

- `hs_initiate_recording` – 定義が完了したことを通知し、レコーディング・セッションの指定開始時刻にセッションを開始できるようにする。
- `hs_terminate_recording` – 進行中のレコーディング・セッションの定義をキャンセルする。

## プレイバック・セッションの開始／終了

`hs_create_playback_session` コマンドは、プレイバックの定義を開始します。このコマンドを発行してから、プレイバック・セッションを定義したり、プレイバックを実行したりする他のコマンドを使用します。

`hs_create_playback_session` コマンドを正常に発行したら、`hs_terminate_playback` コマンドを実行して、プレイバック・セッションを終了します。このコマンドにより、定義がキャンセルされ、プレイバックが終了します。

## Historical Server のコマンド

以降のページでは、Historical Server のコマンドについて説明します。

- [hs\\_create\\_alarm](#)
- [hs\\_create\\_filter](#)
- [hs\\_create\\_playback\\_session](#)
- [hs\\_create\\_playback\\_view](#)
- [hs\\_create\\_recording\\_session](#)
- [hs\\_create\\_view](#)
- [hs\\_delete\\_data](#)
- [hs\\_initiate\\_playback](#)
- [hs\\_initiate\\_recording](#)
- [hs\\_list](#)
- [hs\\_playback\\_sample](#)
- [hs\\_shutdown](#)
- [hs\\_status](#)
- [hs\\_terminate\\_playback](#)
- [hs\\_terminate\\_recording](#)

### hs\_create\_alarm

#### 説明

アラームを作成します。データ項目の値が、指定されたスレッシュホールド値に達すると、アラームがトリガされます。

#### 構文

`hs_create_alarm view_name, data_item_name, data_item_stat, alarm_action, alarm_action_data, alarm_value, occurrence_threshold.`

#### パラメータ

*view\_name*

アラームを適用するデータ項目を含むビューの名前。

*data\_item\_name*

アラームを適用するデータ項目。データ項目に埋め込みスペースが含まれる場合は、その部分を引用符で囲みます。

*data\_item\_stat*

*data\_item\_name* 用の統計タイプ。ビュー定義には、*data\_item\_name* と *data\_item\_stat* が存在していることが必要です。統計タイプは引用符で囲みます。

*alarm\_action*

アラーム状態が発生したときに取るべきアクション。値は次のとおりです。

- **log** – アラーム状態が発生したときにメッセージを記録します。*alarm\_action\_data* パラメータでログ・ファイル名を指定します。
- **execute** – アラーム状態が発生したときにプログラムまたはスクリプト・ファイルを実行します。*alarm\_action\_data* パラメータで、実行するファイルを指定します。

*alarm\_action\_data*

*alarm\_action* を実行するために必要な以下の情報を指定します。

<b>alarm_action</b>	<b>alarm_action_data の内容</b>
log	Historical Server がアラーム・メッセージを記録するファイルの名前。デフォルトは Historical Server のログ・ファイル。存在しないファイルを指定した場合、Historical Server によってそのファイルが作成される。既存のファイルを指定した場合、Historical Server を起動したユーザが、そのファイルに対する書き込みパーミッションを持っている必要がある。
execute	実行するファイルの名前。スペースで区切られたパラメータのリストを指定することもできる。ファイルはすでに存在している必要がある。また、Historical Server を起動したユーザが、そのファイルに対する実行パーミッションを持っている必要がある。

**警告！** アラーム状態が発生し、*alarm\_action* が **execute** である場合、指定されたファイルは、そのアラームを作成したアカウントではなく、Historical Server を起動したアカウントによって実行されます。これは、Historical Server を起動したユーザのアクセス権限が、アラームを定義するユーザからも利用可能であることを意味します。通常はファイルを実行できないユーザも、Historical Server アラームを介すると、そのファイルを実行できる場合があります。

*alarm\_value*

アラームをトリガする値。

*occurrence\_threshold*

Historical Server がスレッショルド・アクションを実行するまでに、スレッショルド状態が発生する回数。

- 例
- 1 この例では、Page I/O という名前のビューに対するアラームを作成します。このビューは、Value for Session 統計タイプの [Page I/O] データ項目を含んでいます。トリガ値が 50 以上になると、*page\_io\_alarm\_file* で指定されたファイルにこれらのデータが記録されます。
- ```
hs_create_alarm PageIO,"Page I/O",
"Value for Session",log,page_io_alarm_file,50
```
- 2 この例では、Page I/O という名前のビューに対するアラームを作成します。[Page I/O] データ項目がトリガ値 100 以上になると、このアラームによって、/user/script1 が実行され、その最初のパラメータとして値 100 が渡されます。
- ```
hs_create_alarm PageIO,"Page I/O",
"Value for Session",execute,
"/user/script1 100",100,6
```
- 使用法
- *alarm\_action* が **execute** の場合、アラーム状態が発生すると、スクリプトやプログラムはバックグラウンドで実行されます。スクリプトやプログラム中のコマンドはフォアグラウンドで実行され、表示が可能な場合には表示も行われます。  
たとえば、アラーム状態においてクロックを表示する場合は、*clock.exe* をスクリプト・ファイルに記述し、アラーム作成時に *clock.exe* を指定するのではなく、そのスクリプト・ファイル名を指定します。
  - アラーム状態が発生し、*alarm\_action* が **log** である場合、Historical Server はログ・ファイルに 3 行書き出します。1 行目の内容は次のとおりです。
    - アラームをトリガしたサンプルのタイムスタンプ
    - アラームが設定されたデータ項目の名前
    - 当該データ項目の統計タイプ
    - アラームをトリガしたデータ項目の値
    - アラームのスレッシュホールド値

• ログ・ファイルの 2 行目は次のとおりです。

    - セッション ID
    - ビュー名
    - アラーム ID

• ログ・ファイルの 3 行目は次のとおりです。

    - モニタしている Adaptive Server の名前
    - アラームをトリガしたサンプル・データのローについての、ビューに含まれるすべてのキー・データ項目の名前と値

- アラーム状態が発生し、*alarm\_action* が **execute** である場合、Historical Server は *alarm\_action\_data* パラメータで指定したスクリプトまたはプログラムを実行します。Historical Server は、*alarm\_action\_data* で指定した引数を使用して、このプログラムまたはファイルを起動します。この引数の後には、プログラム内からアクセスできる追加の位置指定引数が記述されます。Historical Server によって呼び出しに追加される位置指定引数は、以下の情報を示します。
  - セッション ID
  - アラーム ID
  - サンプル・タイムスタンプ
  - アラームが設定されたデータ項目
  - 当該データ項目の統計タイプ
  - アラームをトリガしたデータ項目の値
  - アラームのスレッショルド値
  - モニタしている Adaptive Server の名前
  - ビューにあるキー・データ項目の数
  - キー・データ項目ごとに、以下の情報が含まれます。
    - データ項目名
    - アラームをトリガしたサンプル・データ・ローの、データ項目の値
- アラーム状態が発生し、*alarm\_action* が **execute** である場合、指定したファイルは、そのアラームを作成したアカウントではなく、Historical Server を起動したアカウントによって実行されます。これは、Historical Server を起動したユーザのアクセス権限が、アラームを定義したユーザからも利用可能であることを意味します。

---

**警告！** 通常はファイルを実行できないユーザも、Historical Server アラームを介すると、そのファイルを実行できる場合があります。

---

## hs\_create\_filter

### 説明

データ項目のフィルタリング基準を指定します。フィルタは、ビューにある 1 つのデータ項目についての収集データの範囲を制限します。フィルタの作成はオプションです。

### 構文

```
hs_create_filter view_name, data_item_name, data_item_stat, filter_type,  
value_spec
```

### パラメータ

*view\_name*

フィルタするデータ項目を含むビューの名前。

*data\_item\_name*

フィルタを適用するデータ項目。データ項目に埋め込みスペースが含まれる場合は、その部分を引用符で囲みます。

*data\_item\_stat*

*data\_item\_name* 用の統計タイプ。ビュー定義には、*data\_item\_name* と *data\_item\_stat* が存在していることが必要です。統計タイプは引用符で囲みます。

*filter\_type*

使用するフィルタのタイプ。有効な値は次のとおりです。

- **eq** – 値リストのいずれかに一致する値を通過させます。つまり、値が、1 番目のフィルタ値、2 番目のフィルタ値、リスト中のその他の値のいずれかに一致した場合、値はフィルタを通過します。値のリストは、*value\_spec* パラメータで指定します。
- **neq** – 値リストのいずれにも一致しない値を通過させます。つまり、1 番目のフィルタ値、2 番目のフィルタ値、リスト中の他の値のいずれにも一致しない場合に、値がフィルタを通過します。値のリストは、*value\_spec* パラメータで指定します。
- **range** – 値の範囲内にある値を通過させます。値の範囲は、*value\_spec* パラメータで指定します。
- **top** – サンプル・インターバル中にそのデータ項目について受信した値のうち、大きさで、指定された順位内の値であるものだけを通過させます。たとえば、**top 10** フィルタは、そのデータ項目について大きさで 10 位以内の値を持つサンプル・ローを 10 個通過させます。

*value\_spec*

フィルタによって検査する値を指定します。以下のように、構文は *filter\_type* の値によって異なります。

<i>filter_type</i>	<i>value_spec</i> の構文
eq または neq	<i>value1</i> [, <i>value2</i> ]...
range	<p>ある下限値以上の値を指定するには、次のようにする。</p> <p><i>low,value</i></p> <p>ある上限値以下の値を指定するには、次のようにする。</p> <p><i>high,value</i></p> <p>ある下限値以上で、ある上限値以下の値を指定するには、次のようにする。</p> <p><i>low,value1,high,value2</i></p>
top	<p>[Object Name] データ項目と [Procedure Name] データ項目の値の指定については、「<a href="#">オブジェクト名とプロシージャ名に関するフィルタの指定</a>」(54 ページ) を参照。これらのデータ項目のいずれかに上限値と下限値の両方を指定する場合、各限界値には同数のサブコンポーネントが必要。</p> <p>値</p> <p>ここで、<i>value</i> は通過する項目の数。0 より大きくなければならない。</p>

## 例

この例では、Page I/O という名前のビュー用のフィルタを作成します。このビューは、“Value for Session”統計タイプの [Page I/O] データ項目を含んでいます。この例では、サンプル用に記録されるローを、現在までのレコーディング・セッションの Page I/O の総数で最大となった 20 個のローに制限します。

```
hs_create_filter PageIO,"Page I/O","Value for Session",top,20
```

## 使用法

## ビューでの複数フィルタの使用

フィルタは Historical Server が記録するデータ量を制限します。各フィルタはビュー内のデータ項目 1 つに適用されます。各データ項目は、その項目に適用されるフィルタを 1 つだけ持つことができます。

ビューの複数のデータ項目にフィルタが存在する場合、Historical Server はすべてのフィルタを満たすデータを記録します。実際には、プール AND 演算子はさまざまなフィルタの値に適用されます。

あるデータ項目に **top** フィルタが機能し、他のデータ項目に別のフィルタが機能している場合、**top** フィルタを通過した一部の値は他のフィルタによって排除される場合があります。結果は、**top** よりも少なくなり、*n* 個の値が返されます。

### ワイルドカードの使用

ワイルドカード文字はパーセント記号 (%) です。*filter\_type* が **eq** または **neq** の場合、文字列を返すデータ項目の *value\_spec* にワイルドカードを使用できます。ワイルドカードは、0 文字以上のあらゆる文字列に一致します。ワイルドカードはフィルタ値内のあらゆる場所(先頭、末尾、中間のあらゆる場所)に使用できます。*value\_spec* には、ワイルドカードを 1 つだけ使用できます。

ワイルドカードの单一インスタンス規則の例外は、[Object Name] データ項目と [Procedure Name] データ項目です。それらのマルチコンポーネントのデータ項目では、コンポーネントごとにワイルドカードのインスタンスを 1 つ使用できます。詳細については、「[オブジェクト名とプロシージャ名に関するフィルタの指定](#) (54 ページ)」を参照してください。

*filter\_type* が **range** の場合、*value\_spec* にワイルドカード文字を使用することはできません。

### オブジェクト名とプロシージャ名に関するフィルタの指定

*filter\_type* が、**eq**、**neq**、**range** の場合、[Object Name] データ項目と [Procedure Name] データ項目は、*value\_spec* でマルチコンポーネントの値を使用することができます。これらの複合フィルタ値は、以下の名前の形式を使用することができ、その場合、*object\_name* は以下のようにテーブル名かストアド・プロシージャ名になります。

- *object\_name*
- *owner\_name.object\_name*
- *database\_name.owner\_name.object\_name*

*filter\_type* が **eq** または **neq** の場合、各コンポーネントはワイルドカード文字 (%) の单一のインスタンスを含むことができます。ワイルドカードは、コンポーネント全体の代わりに使用することができます。たとえば、*data\_item* Object Name の %.%.authors の *value\_spec* は、所有者が誰であるかに関係なく、すべてのデータベースの authors テーブルを返すフィルタを設定します。ワイルドカード文字の詳細については、「[ワイルドカードの使用](#) (54 ページ)」を参照してください。

*filter\_type* が **range** で上限と下限の両方が設定されている場合、この 2 つの限界値は同じ数のサブコンポーネントを持っている必要があります。

コンポーネント・データ項目を使用するフィルタを作成する場合には、次のような考慮事項があります。

- “Object Name” データ項目に対する複合フィルタの値に、*database\_name* が含まれる場合、そのビューに “Database Name” データ項目も存在していること。
- “Object Name” データ項目に対する複合フィルタの値に、*owner\_name* が含まれる場合、そのビューに “Owner Name” データ項目も存在していること。
- “Procedure Name” データ項目に対する複合フィルタ値の値に、*database\_name* が含まれる場合、そのビューに “Procedure Name” データ項目も存在していること。
- “Procedure Name” データ項目に対する複合フィルタの値に、*owner\_name* が含まれる場合、そのビューに “Procedure Owner Name” データ項目も存在していること。

## hs\_create\_playback\_session

### 説明

プレイバック・セッションの特性を定義します。このコマンドはプレイバック・セッションを作成する際の第一歩となるものです。

### 構文

```
hs_create_playback_session start_time, end_time, summarization_interval,
allow_estimation, missing_data_option, target, directory_name, protection_level,
sample_interval, script_type, delete_option, session_id [, session_id...]
```

### パラメータ

#### start\_time

記録されたデータがプレイバックされ始める日付と時刻を指定します。デフォルトでは、*session\_id* で指定された最初のセッションの開始からプレイバックを始めるようになっています。デフォルト設定を使用するには NULL 値を使用します。

*start\_time* のフォーマットは次のとおりです。

“year/month/day hour:minute[:second] [time zone]”

*summarization\_interval* パラメータと *missing\_data\_option* パラメータの値、および指定時刻に使用できるデータが存在するかどうかによって、プレイバックでは指定時刻よりも遅い時刻のデータを使用する場合があります。ただし、*start\_time* よりも早い時刻のデータを使用することはありません。

*start\_time* を指定する場合、*session\_id* で指定されたセッションのうち、少なくとも 1 つのセッションの終了時刻よりも早い時刻であることが必要です。次のタイム・ゾーン・オプションを使用できます。

パラメータ値	説明
EST	米国東部標準時。
EDT	米国東部夏時間。
CST	米国中部標準時。
CDT	米国中部夏時間。
MST	米国山地標準時。
MDT	米国山地夏時間。
PST	米国太平洋標準時。
PDT	米国太平洋夏時間。
MET	中央ヨーロッパ標準時。
MET DST	中央ヨーロッパ夏時間。
WET DST	西ヨーロッパ(グリニッジ)夏時間。
GMT	グリニッジ標準時。夏時間に関係なく、西ヨーロッパ(グリニッジ)タイム・ゾーンと同じ。標準時または夏時間が有効になっているときには、EST、EDT など前述のタイム・ゾーンの指定は、日付と時刻の組み合わせの形でのみ行うことができる。GMT はすべての日付および時刻指定と組み合わせができる。
GMT{+-} <i>hours_offset</i>	その他のタイム・ゾーンを指定する場合、 <i>hours_offset</i> に、現地時間を算出するためにグリニッジ標準時に加える時間数を指定する。オフセット値の許容範囲は -24 ~ +24 時間。+5.5 のような小数点以下を含むオフセットも使用できる。

デフォルトの *time\_zone* は、Historical Server のローカル・タイム・ゾーンです。

#### *end\_time*

プレイバックされる記録されたデータが終了する日付と時刻を指定します。デフォルトでは、指定された最後のセッションの終了時にプレイバックも終了するようになっています。デフォルト設定を使用するには NULL 値を使用します。

このパラメータのフォーマットは、*start\_time* のフォーマットと同じです。*end\_time* を指定する場合、少なくとも 1 つの入力セッションの *start\_time* よりも遅い時刻であることが必要です。

*summarization\_interval*

プレイバックの詳細なレベルを指定する必須パラメータ (デフォルト値はありません)。有効な値は次のとおりです。

- **raw** — 収集されたデータを、同じサンプル・インターバルでプレイバックします。ロー・データを記録されたとおりに表示する場合は、このオプションを選択します。また、このオプションは、現在の SQL 文データ、ロックやプロセスのステータス情報などのスナップショット・データをプレイバックするために利用可能な唯一のオプションです。スナップショット・データ項目の一覧については、[表 C-2 \(160 ページ\)](#) を参照してください (スナップショット・データ項目は、「raw 以外に使用可」カラムが「いいえ」の項目です)。

このオプションは、*target* が **client** の場合のみ有効です。

- **actual** — 入力レコーディング・セッションと同じサンプル・インターバルでデータをプレイバックします。

このオプションを使用すると、プレイバック・ビューで一部のデータ項目変更を指定することができます。また、Historical Server は、レコーディング・セッション時間が、要求されたプレイバック・セッション時間と合わない場合、最初と最後のサンプルに対して必要な調整を行います。

要約の必要がなく、特定のデータ項目を追加または変更する場合は、このオプションを選択します。また、*target* が **file** に設定されている場合、このオプションを使用すると、複数レコーディング・セッションの非スナップショット・データを結合することができます。

このオプションは、*target* が **client** の場合のみ有効です。

- **entire** — 単一サンプルとして要約された各入力レコーディング・セッション用のデータをプレイバックします。サンプル間隔は、要求されたプレイバックの *start\_time* から *end\_time* までの時間となります。

このオプションを使用すると、プレイバック・ビューで一部のデータ項目変更を指定することができます。また、Historical Server は、要求されたプレイバック・セッションの開始と終了時刻を正確に反映させるため、データ値に必要な調整を行います。

記録したデータを統合して、長期間にわたるアクティビティの詳細を概要情報にまとめる場合は、このオプションを選択します。

*sample\_interval*

指定した長さのサンプル・インターバルに要約されたデータを Playback します。このパラメータ値はサンプル・インターバルの長さであり、以下のように指定されます。

"S"  
"M:S"  
"H:M:S"  
"D H:M:S"

パラメータの意味は次のとおりです。

- *S* – 秒
- *M* – 分
- *H* – 時間
- *D* – 日

すべてのコンポーネントは 1 桁または 2 桁の数値です。以下は、その例です。

"30" (specifies sample intervals of 30 seconds)  
"10:0" (specifies sample intervals of 10 minutes)  
"8:30:0" (specifies sample intervals of 8 1/2 hours)  
"5 0:0:0" (specifies sample intervals of 5 days)

最初のサンプル間隔は *start\_time* に開始し、最後のサンプルを除くすべてのサンプルが指定された長さを持ちます。

このオプションを使用すると、Playback ビューで一部のデータ項目変更を指定することができます。また、Historical Server は、要求された Playback ビュー長と Playback サンプル・インターバルを正確に反映させるため、必要な調整をデータ値に対して行います。

データを任意の細分性で要約する場合は、このオプションを選択します。この要約タイプでは、アクティビティの偏りを調整できるので、長期的な傾向を調べるときに便利です。

**actual**、**entire**、ユーザ定義の間隔の各オプションには、以下のような共通の特徴があります。

- プレイバック・ビューにおけるデータ項目の違い – 3 つのすべてのオプションで、プレイバック・ビューはインプット・ビューのものとは異なる統計タイプを使用できます。また、インプット・ビューになかったいくつかの推定データ項目や計算データ項目を含めることもできます。プレイバック・ビューには、現在の SQL 文データのようなスナップショット・データを含めることはできません。これらのオプションの選択時にプレイバック・ビューに含めることのできるデータ項目と統計タイプのリストについては、[表 C-2 \(160 ページ\)](#) を参照してください。

- データ調整 – 3つのすべてのオプションで、Historical Server は、以下の差異を補正するために必要な数値計算を実行します。
  - 入力レコーディング・セッション・サンプル・インターバルと要求されたプレイバック・サンプル・インターバル
  - 入力レコーディング・セッションのセッション長と要求されたプレイバックの開始時刻と終了時刻

これらの差異を処理するため、累積数の配分が行われます。パーセンテージや比率は加重平均化され、各入力サンプルが寄与する秒数によって加重されます。

詳細については、「[付録 C プレイバック・ビューの定義の仕様](#)」を参照してください。

#### *allow\_estimation*

正確な計算が不可能なデータ項目に対する値をプレイバックに推定させるかどうかを指定します。有効な値は次のとおりです。

- disallow** (デフォルト) – 推定を必要とするデータ項目を検出した場合に、`hs_create_playback_view` コマンドがエラーを返すように設定します。
- allow** – 必要に応じ、プレイバックがデータ項目を推定するように設定します。プレイバック・セッションで推定を許可しない限り、一部のデータ項目はプレイバック・ビューに含めることはできません。

*summarization\_interval* が `raw` の場合、このパラメータは無視されます。

推定を要求するデータ項目を確認するには、[表 C-2 \(160 ページ\)](#) を参照してください。

#### *missing\_data\_option*

*target* が `client` の場合、このパラメータは、入力セッションで利用できるデータがないときに `hs_playback_sample` コマンドで時間をどのように処理するかを指定します。有効な値は次のとおりです。

- skip** (デフォルト) – ある期間のデータが存在しない場合、`hs_playback_sample` コマンドはデータが存在しないサンプルを返すのではなく、データが存在する次の期間に直接移行します。
- show** – `hs_playback_sample` コマンドは、利用できるデータがない時間についてもサンプルを返します。クライアント・プレイバックの間、ゼロ個のローとともにカラム・ヘッダが返されます。

*target* が `file` の場合、ギャップは許可されません。

**ターゲット**

プレイバック・セッションのターゲットとなる結果を指定します。有効な値は次のとおりです。

- **client** (デフォルト) – クライアントがプレイバックを利用できるようにします。
- **file** – 現在の `hs_create_playback_session` コマンドと後続の `hs_create_playback_view` コマンドによって指定されたすべてのデータを含む新しいセッションを作成できるようにします。新しいセッションを作成するには、`hs_initiate_playback` コマンドを使用します。`summarization_interval` パラメータが `raw` または `actual` の場合、`target` を `file` に設定することはできません。

*directory\_name*

*target* が `file` の場合、このパラメータは新しいファイルが作成されるディレクトリを指定します。このパラメータは、*target* が `client` のときは無視されます。デフォルトは Historical Server のホーム・ディレクトリです。

Historical Server を起動するオペレーティング・システム・アカウントは、指定したディレクトリに対して実行 (検索) と書き込みパーミッションを持っていなければなりません。

*protection\_level*

*target* が `file` の場合、このパラメータは、`hs_initiate_playback` コマンドが新しく作成されたセッションのデータ・ファイルを割り当てるパーミッション・レベルを指定します。この保護レベルは、(`hs_list` コマンドを使用した) 制御ファイルのメタデータの閲覧と、(`hs_create_playback_session` コマンドを使用した) データでのプレイバック・セッションの作成を制御します。有効な値は次のとおりです。

- **private** (デフォルト) – 新しいファイルをパスワードで保護するよう指定する。ファイルは、ファイルを作成したものと同じアカウントか、Historical Server のスーパーユーザからのみアクセスできる。
- **public** – 新しいセッションのファイルのデータに対して制限なしでアクセスできるようにする。
- **null** – デフォルト値の `private` とみなす。

*target* が `client` のときは、このパラメータは無視されます。

*script\_type*

*target* が *file* の場合、このパラメータは、*hs\_initiate\_playback* コマンドによって、新しく作成されたセッションのスクリプト・ファイルを作成するかどうかを指定します。スクリプト・ファイルには、新しいセッション用に定義された各プレイバック・ビューの Adaptive Server テーブルを作成する SQL コマンドが含まれます。有効な値は次のとおりです。

- **no\_script** (デフォルト) – 新しいセッション用にスクリプト・ファイルを作成しない。
- **sybase\_script** – スクリプト・ファイルを作成する。ファイルは Historical Server のホーム・ディレクトリに保存される。名前は *sSessionId*。
  - *s* は定数。
  - *SessionId* は Historical Server によって割り当てられる。
- **null** – デフォルト値の *no\_script* を実装する。

*target* が *client* のときは、このパラメータは無視されます。

*delete\_option*

*target* が *file* の場合、このパラメータは、*hs\_initiate\_playback* コマンドが新しいセッションを正常に作成した後に、入力セッション・ファイルを削除するかどうかを指定します。有効な値は次のとおりです。

- **retain** (デフォルト) – 入力セッション用のデータを含むファイルを削除しない。
- **delete** – 新しいセッションが正常に作成された場合、すべての入力セッション・ファイルを削除する。ユーザは、Historical Server のスーパーユーザか、すべての入力セッションの所有者のいずれかでなければならぬ。
- **null** – デフォルト値の *retain* とみなす。

*target* が *client* のときは、このパラメータは無視されます。

*session\_id[,session\_id...]*

入力セッションをプレイバックするためのユニークな識別子を指定します。少なくとも 1 つの *session\_id* が必要です。複数の *session\_id* が使用されている場合、それらの開始時刻に従ってそれらを正しい順序で指定する必要があります。最も早い開始時刻を最初に指定します。プレイバックでは、順序が正しくないセッションはスキップされます。 *summarization\_interval* が *raw* の場合、有効な *session\_id* は 1 つだけです。

Historical Server は、レコーディング・セッションが定義されたときに、セッション ID をレコーディング・セッションに割り当てます。プレイバックするセッション ID を検索するには、*hs\_list* コマンドを使用します。

*target* が **file** の場合、入力セッションの終了時刻と次の入力セッションの開始時刻が同じになるように指定する必要があります。また、指定される各セッションはレコーディングを完了していなければなりません。

*target* が **client** の場合、各セッションでレコーディングが開始されている必要がありますが、**hs\_create\_playback\_session** コマンドを送信すると、レコーディングを続行できます。

### 例

- 1 この例では、単一の入力セッション(セッション 7)に基づき、プレイバック・セッションを作成します。セッションは、要約、正規化や新しいセッションの作成が行われることなく、全体がプレイバックされます。

```
hs_create_playback_session null, null, raw, null,  
null, client, null, null, null, null, 7
```

- 2 この例では、3つの入力セッション(セッション 4、6、9)に基づき、プレイバック・セッションを作成します。9:00～5:00までの時間帯に収集されたデータのみがプレイバックされます。プレイバックされるデータは、30分間隔で要約されます。このプレイバックは新しいセッションとして保存されます。

```
hs_create_playback_session "1996/5/3 9:00", "1996/5/3  
17:00", "0 00:30:00", disallow, skip, file, null,  
public, no_script, retain, 4, 6, 9
```

### 使用法

- Historical Serverへの接続は、レコーディング・セッションの定義と、プレイバック・セッションの定義／実行のいずれか一方を行うことができますが、両方を行うことはできません。  
*hs\_create\_playback\_session* コマンドが正常に実行されたら、*hs\_terminate\_playback* コマンドを正常に実行してから、レコーディング・セッションの定義を開始するか、別のプレイバック・セッションの定義を開始します。
- プレイバック・セッションの動作は、*target* パラメータの値により異なります。
  - target* が **client** の場合、クライアントは *hs\_playback\_sample* コマンドを使用してデータを取得します。プレイバックは、まだレコーディング処理中のレコーディング・セッションから実行することもできます。
  - target* が **file** の場合、クライアントはこのプレイバック・セッションからデータを取得しません。代わりに、Historical Server が新しいセッションを作成し、その新しいセッションのデータ・ファイルにこのプレイバック・セッションのデータを入力します。新しいセッションへのアクセスの容易さは、*protection\_level* パラメータによって決まります。このコマンドを呼び出すユーザは、新しいセッションの所有者です。*hs\_initiate\_playback* コマンドは新しいセッションを作成します。

プレイバックは、`hs_create_playback_session` コマンドを送信した時点でレコーディングを終了しているセッションに限定されます。

- `target` が `file` で複数のプレイバック・セッションが入力として使用されている場合、時刻のギャップは許可されません。たとえば、月曜日から金曜日まで毎日、午前 9 時から午後 5 時までデータを収集する場合、これらの 5 つのレコーディング・セッションをプレイバックして、新しく要約された週のセッションを作成することはできません。ただし、月曜日から金曜日まで毎日、午前 9 時から午前 9 時までデータを収集して空白時間を解消すれば、プレイバック機能を使用して新しく要約された週のセッションを作成できます。午前 9 時から午後 5 時までのレコーディング・セッションを維持し、これとは別に午後 5 時から午前 9 時までのレコーディング・セッションを追加して、空白時間を解消する方法もあります。オフ時間のサンプル間隔を長くすると、収集するデータの量が減ります。
- `hs_create_playback_session` コマンドで使用される `summarization_interval` パラメータの値は、`hs_create_playback_view` コマンドがビューを定義するときに使用する規則に影響します。

## hs\_create\_playback\_view

### 説明

プレイバック・セッションに含める入力レコーディング・セッションのビューを指定します。また、プレイバックする各ビューのデータ項目を指定します。

### 構文

```
hs_create_playback_view view_name,  
[data_item_name_1, data_item_stat_1  
[.data_item_name_2, data_item_stat_2]...]
```

### パラメータ

#### view\_name

プレイバックするビューの名前。プレイバック・ビューの名前は、入力セッションの対応するビューの名前と一致している必要があります。2 つ以上の入力セッションが存在する場合、そのビューがすべての入力セッションに存在していること、どの入力セッションにもまったく同じデータ項目とフィルタが含まれていることが必要です。

特定のビュー名に対して定義できるプレイバック・ビューは 1 つだけです。別のプレイバック・ビューと同じ名前で作成しようとすると、エラーになります。

入力セッションにビューをリストし、各ビューにデータ項目をリストするには、`hs_list` コマンドを使用します。

*data\_item\_name\_n*

プレイバック・ビューで表示させる入力ビューに存在するデータ項目(入力ビューになかったデータ項目を表示できる場合もあります)。データ項目に埋め込みスペースが含まれる場合は、その部分を引用符で囲みます。

データ項目をまったく指定しない場合、ビューは、入力セッションの対応するビューのすべてのデータ項目を使用して定義されます。しかし、この対応するビューからのデータ項目がプレイバック・ビューで有効でない場合、エラーが発生します。

プレイバック・ビューの設計の詳細については、[表 C-2 \(160 ページ\)](#) を参照してください。

*data\_item\_stat\_n*

各データ項目の統計タイプ。統計タイプは、入力ビューで使用されている統計タイプと必ずしも同じである必要はありません。プレイバック・ビューのデータ項目の有効な統計タイプを確認するには、[表 C-2 \(160 ページ\)](#) を参照してください。

統計タイプは引用符で囲みます。

例

- 1 この例では、入力セッションのビュー *device\_view* に基づくプレイバック・ビューを作成します。プレイバック・ビューには、入力ビューに存在するすべてのデータ項目が含まれます。データ項目と統計タイプも入力ビューと同じ組み合わせです。

```
hs_create_playback_view device_view
```

- 2 この例では、入力セッションからタイムスタンプと 2 つのデータ項目を返すプレイバック・ビューを作成します。

```
hs_create_playback_view
device_view,"Timestamp","Value for Sample","Device
Name", "Value for Sample", "Device Reads","Rate for
Sample"
```

使用法

- *hs\_create\_playback\_view* コマンドはプレイバック・セッションの定義中にのみ有効です。つまり、このコマンドは、*hs\_create\_playback\_session* コマンドが発行されてから *hs\_initiate\_playback* コマンドが発行されるまでの間に発行する必要があります。
- プレイバック・セッションは少なくとも 1 つのビューを保持している必要があります。
- 「[付録 C プレイバック・ビューの定義の仕様](#)」に、有効なプレイバック・ビューを設計するための要件を示します。これには、プレイバック・ビューに対して有効なデータ項目と統計タイプの組み合わせも記載されています。

## hs\_create\_recording\_session

### 説明

新しいレコーディング・セッションを確立します。

### 構文

```
hs_create_recording_session monServerName, sample_interval
  [,dir_name][,start_time][,end_time]
  [,protection_level][,error_option][,script_type], [tab_delimited]
```

### パラメータ

*monServerName*  
履歴データの収集対象とする Adaptive Server からデータを収集するために使用する Monitor Server の名前。Historical Server に接続するために使用するユーザ名とパスワードは、ここで指定する Monitor Server によりモニタされる Adaptive Server 上の有効なログイン・アカウントと一致している必要があります。

### *sample\_interval*

サンプル相互間の時間を秒数で指定します。有効なサンプル・インターバルは最短で 1 秒、最長で 86400 秒(1 日の秒数)です。

### *dir\_name*

レコーディング・セッションの結果作成される履歴データ・ファイルが保存されるディレクトリのパス名。Historical Server を起動するオペレーティング・システム・アカウントは、このディレクトリに対して実行(検索)と書き込みパーミッションを持っていなければなりません。

デフォルトは Historical Server のホーム・ディレクトリです。ホーム・ディレクトリは、Historical Server の起動コマンドの -D パラメータで指定されます。

### *start\_time*

レコーディングが開始される予定の日付と時刻であり、次のフォーマットを使用します。

```
year/month/day hour:minute[:second] [time zone]
```

デフォルトでは、すぐに記録を開始します。次のタイム・ゾーン・オプションを使用できます。

パラメータ値	説明
EST	米国東部標準時。
EDT	米国東部夏時間。
CST	米国中部標準時。
CDT	米国中部夏時間。
MST	米国山地標準時。
MDT	米国山地夏時間。
PST	米国太平洋標準時。
PDT	米国太平洋夏時間。
-T	中央ヨーロッパ標準時。
MET DST	中央ヨーロッパ夏時間。
WET DST	西ヨーロッパ(グリニッジ)夏時間。

パラメータ値	説明
GMT	グリニッジ標準時。夏時間に関係なく、西ヨーロッパ(グリニッジ)タイム・ゾーンと同じ。標準時または夏時間が有効になっているときには、EST, EDT など前述のタイム・ゾーンの指定は、日付と時刻の組み合わせの形でのみ行うことができる。GMT はすべての日付および時刻指定と組み合わせることができる。
GMT{+-} <i>hours_offset</i>	その他のタイム・ゾーンを指定する場合、 <i>hours_offset</i> に、現地時間を作成するためにグリニッジ標準時に加える時間数を指定する。オフセット値の許容範囲は -24 ~ +24 時間。 +5.5 のような小数点以下を含むオフセットも使用できる。

タイム・ゾーンを指定しない場合、Historical Server に設定されているタイム・ゾーンが使用されます。

*start\_time* には、現在の時刻から 31 日以上先の時刻を指定することはできません。

*end\_time*

レコーディングを停止する時刻。デフォルトでは、*start\_time* の 24 時間に後でレコーディング・セッションを停止します。*end\_time* を指定し、*start\_time* を指定していない場合、*start\_time* はデフォルトで現在の時刻に設定されます。

*protection\_level*

レコーディング・セッションで作成されたデータ・ファイルをパスワードで保護するか、すべてのユーザにアクセス可能にするかを指定します。この保護レベルは、(hs\_list コマンドを使用した) 制御ファイルのメタデータの閲覧と、(hs\_create\_playback\_session コマンドを使用した) データでのプレイバック・セッションの作成を制御します。

- **private** (デフォルト) – レコーディングされたデータをパスワードで保護するよう指定する。ファイルは、ファイルを作成したものと同じアカウントか、Historical Server のスーパーユーザからのみアクセスできます。
- **public** – レコーディングされたデータに対して制限なしでアクセスするよう指定する。

*error\_option*

Historical Server によって、レコーディング・セッション中の致命的でないエラーをどのように処理させるかを指定します。有効な値は次のとおりです。

- **continue** (デフォルト) – 致命的でないエラーが検出された場合にレコーディングを続行するよう指定する。たとえば、モニタリング情報を要約中に Monitor Server による書き込みで指定されたバッファが一杯になった場合、1つ以上の致命的でないエラーが Historical Server に送信される。しかし Historical Server は引き続き、現在のサンプルや今後のサンプルから利用可能な情報を収集することができる。
- **halt** – 致命的でないエラーが検出された場合にレコーディング・セッションを終了するよう指定する。

*script\_type*

ユーザのレコーディング・セッション・ビュー定義からテーブルを作成するスクリプト・ファイルを、Historical Server によって作成するかどうかを指定します。

- **sybase\_script** – スクリプト・ファイルを作成する。スクリプト・ファイルには、レコーディング・セッションのビューごとに Adaptive Server テーブルを作成する SQL コマンドが含まれる。
- **no\_script** (デフォルト) – スクリプト・ファイルを作成しない。

*tab\_delimited*

**hs\_create\_recording\_session** にこのパラメータを追加すると、出力ファイル・デリミタに HS を使用するデフォルトが無視されます。代わりに TAB 文字が使用されます。

## 例

この例では、SERVER1\_MON という名前の Monitor Server からデータを取得するためのレコーディング・セッションを作成します。データは 30 秒ごとに取得され、/user/hist\_dir ディレクトリのデータ・ファイルに書き込まれます。開始は即時で、終了は 1997 年 8 月 8 日午前 10 時 30 分 (米国東部夏時間) です。生成されるファイルは private の制限を持ちます。致命的でないエラーが検出された場合でもレコーディング・セッションは継続します。記録された各ビュー用の Adaptive Server テーブルを作成するスクリプト・ファイルを作成します。

```
hs_create_recording_session SERVER1_MON,30,
    /user/hist_dir,NULL,"97/08/08 10:30 EDT",
    private,continue, sybase_script
```

## 使用法

- 致命的なエラーには、クラッシュや、その他の Adaptive Server または Monitor Server との接続不良が含まれます。致命的なエラーが発生すると、*error\_option* の値に関係なく、レコーディング・セッションが強制終了されます。

- ユーザは、このコマンドを発行してから、レコーディング・セッションのビュー、アラーム、またはフィルタを定義します。このコマンドを発行し、少なくとも 1 つのビューを定義してから、hs\_initiate\_recording コマンドを使用してレコーディングを開始します。

## hs\_create\_view

### 説明

セッション中に記録するデータ項目のコレクションを定義します。

### 構文

```
hs_create_view view_name,  
  data_item_name_1, data_item_stat_1  
  [,data_item_name_2, data_item_stat_2]...
```

### パラメータ

#### view\_name

ユーザ定義のビュー名。ビュー名は a – z、A – Z、0 – 9 の文字とアンダースコア (\_) で構成されている必要があります。

#### data\_item\_name\_n

ビューに表示されるデータ項目。データ項目に埋め込みスペースが含まれる場合は、その部分を引用符で囲みます。

利用可能なデータ項目のリストについては、[表 A-1 \(103 ページ\)](#) を参照してください。ビュー内のデータ項目の有効な組み合わせについては、[表 B-2 \(122 ページ\)](#) を参照してください。

#### data\_item\_stat\_n

各データ項目の統計タイプ。統計タイプは引用符で囲まれていなければなりません。有効な統計タイプは次のとおりです。

- “Value for Sample”
- “Value for Session”
- “Rate for Sample”
- “Rate for Session”
- “Avg for Sample”
- “Avg for Session”

データ項目によって、有効な統計タイプと無効な統計タイプがあります。詳細については、[表 B-3 \(145 ページ\)](#) を参照してください。

### 例

この例では、Value for Session 統計タイプの [Page I/O] データ項目と、Value for Sample 統計タイプの [Kernel Process ID] データ項目を含む、Page I/O という名前のビューを作成します。

```
hs_create_view PageIO,"Page I/O","Value for  
Session","Kernel Process ID","Value for Sample"
```

## 使用法

- ビュー用の記録データ・ファイルにデータ項目が保存される順序は、ユーザがオプションを指定した順序によって決定されます。
- 各レコーディング・セッションには、少なくとも 1 つのビューを定義しなければなりません。
- Historical Server のプレイバック機能を利用してレコーディング・セッション中のデータを検査する場合、レコーディング・セッション・ビューを設計しながら、プレイバック・ビューについて検討することもできます。

多くのデータ項目について、Historical Server は、レコーディング中に使用したものとは異なる統計タイプを使用してデータ項目をプレイバックすることができます。しかし、一部のデータ項目については、ある追加のデータ項目が記録データに含まれない限り、Historical Server は（正確な計算を行うのではなく）値を推定することでデータの要約を行います。正確な計算を行うために必要なデータ項目を確認するには、[表 C-2 \(160 ページ\)](#) を参照してください。

**hs\_delete\_data**

## 説明

1 つ以上の非アクティブなレコーディング・セッションに関連する履歴モニタリング・ファイルを削除します。

## 構文

`hs_delete_data low_session_id [,high_session_id]`

*low\_session\_id*

データが削除される最初のレコーディング・セッションのユニークな識別子。

*high\_session\_id*

データが削除されるセッションの範囲にある最後のセッションのユニークな識別子。省略した場合、*low\_session\_id* で識別されるセッションのデータだけが削除されます。

## 例

この例では、1 ~ 15 のセッション ID を持つ非アクティブなレコーディング・セッションの履歴モニタリング・ファイルを削除します。

```
hs_delete_data 1,15
```

## 使用法

- 履歴データ・ファイルを削除することができるるのは、以下のユーザのみです。
  - Historical Server のスーパーユーザ
  - レコーディング・セッションが定義されたときと同じユーザ名とパスワードを使用して Historical Server に接続しているユーザ

## hs\_initiate\_playback

---

- 非アクティブなレコーディング・セッション・ファイルのみが削除されます。アクティブなレコーディング・セッションに関するファイルは、そのセッション ID が `hs_delete` コマンドで指定された範囲内にある場合でも削除されません。
- 非アクティブなレコーディング・セッションとは、`end_time` に達したり、`hs_terminate_recording` コマンドが発行されたりしたために終了したセッションです。

## **hs\_initiate\_playback**

### 説明

プレイバック・セッションの定義が終了し、プレイバックを実行する準備が整ったことを指定します。プレイバック・ターゲットが `file` である場合、このコマンドはプレイバックを開始します。プレイバック・ターゲットが `client` の場合、このコマンドはプレイバックを初期化し、デフォルト値の `step` が有効になっている `hs_playback_sample` コマンドがプレイバック・セッションで最初のサンプルを返すようにします。

### 構文

`hs_initiate_playback`

### 使用法

- `hs_create_playback_session` コマンドと正常に発行された 1 つ以上の `hs_create_playback_view` コマンドが、`hs_initiate_playback` コマンドより前に発行されている必要があります。
- `hs_initiate_playback` コマンドの発行前にビューが作成されていない場合、`hs_initiate_playback` はエラーを返します。エラーが返された場合でも、ユーザは `hs_create_playback_view` コマンドを実行したり、`hs_terminate_playback` コマンドでプレイバック・セッションをキャンセルしたりできます。

## **hs\_initiate\_recording**

### 説明

レコーディング・セッションの定義が完了したことを指定し、そのレコーディングの開始、またはレコーディング・セッションの `start_time` に開始するスケジューリングを要求します。

### 構文

`hs_initiate_recording`

### 使用法

- `hs_initiate_recording` コマンドを発行する前に、`hs_create_recording_session` コマンドと 1 つ以上の `hs_create_view` コマンドが発行されている必要があります。
- 1 つ以上の `hs_create_alarm` コマンドと `hs_create_filter` コマンドも、`hs_initiate_recording` コマンドより前に発行できます。

- `hs_initiate_recording` コマンドを実行後、そのセッションを後からキャンセルすることはできますが、このレコーディング・セッションの記述に、ビュー、フィルタ、アラームをさらに追加することはできません。

## hs\_list

### 説明

過去から現在までのレコーディング・セッションについての情報をリストします。

### 構文

`hs_list level [,restriction]`

#### *level*

各レコーディング・セッション用に返された詳細レベルを指定します。指定できる値は、`sessions`、`views`、`data_items`、`alarms`、`filters`、`summarization_data_items` です。

#### *restriction*

現在の *level* 用のレコーディング・セッションについて利用可能なデータのサブセットを選択します。

<b>level</b>	<b>restriction の対象</b>
<code>sessions</code>	<ul style="list-style-type: none"> <li>• 値は次のいずれかになる。           <ul style="list-style-type: none"> <li>• <code>active</code> – 返される値を、この Historical Server インスタンスの現在アクティブなレコーディング・セッションに制限する</li> <li>• <code>inactive</code> – 完了したレコーディング・セッションのみを選択する</li> <li>• <code>latest</code> – (該当するものがある場合) 現在のクライアント接続で最も最近に開始されたレコーディング・セッションのみを選択する</li> </ul> </li> <li>• <i>restriction</i> が省略された場合、ユーザがアクセスできるパーミッションを持っているアクティブ、非アクティブのあらゆるレコーディング・セッションがリストされる。</li> </ul>
<code>views</code>	<ul style="list-style-type: none"> <li>• リストするビューを持つセッション ID。</li> <li>• <i>restriction</i> が省略された場合、ユーザがアクセスできるパーミッションを持っているすべてのレコーディング・セッションのビューがリストされる。</li> </ul>

level	restriction の対象
data_items、 alarms、filters、 または summarization — data_items	<ul style="list-style-type: none"> <li>セッション ID。リストするデータ項目を持つビュー名も指定できる。構文は次のとおり。</li> </ul> <p style="text-align: center;"><b>session_id [,view_name]</b></p> <p>session_id は、単一セッションに含まれるデータ項目にリストを限定する。view_name ( 指定された場合 ) は、そのセッションに対して定義された単一ビューに表示されるデータ項目にリストを限定する。</p> <ul style="list-style-type: none"> <li>restriction が省略された場合、ユーザがアクセスできるパミッショントを持っているアクティブ、非アクティブのあらゆるレコーディング・セッションに対して定義されたすべてのビューに含まれるデータ項目がリストされる。</li> </ul>

## 例

- この例では、Historical Server のこのインスタンスについて、すべてのアクティブなセッションをリストします。

```
hs_list sessions,active
```

- この例では、ID が 10 であるセッションの Page I/O ビューのデータ項目に対して定義されているすべてのアラームをリストします。

```
hs_list alarms,10,PageIO
```

## 使用法

- hs\_list コマンドは、integer データ型として以下のフィールドを返します。
  - セッション ID
  - サンプル・インターバル
  - アラーム数
  - フィルタ数
- hs\_list コマンドは、float データ型として以下のフィールドを返します。
  - アラーム値
- すべての他のフィールドは文字列として返されます。
- セッションを private 保護レベルで記録し、現在のユーザが Historical Server のスーパーユーザでない場合、現在のユーザの名前とパスワードは、そのセッションが表示される前にセッションを記録したユーザの名前とパスワードと照合されます。

### 返されるデータの説明

`hs_list` が返すデータは、*level* の値によって異なります。

- `sessions` は、セッションごとに 1 つのローを以下のカラムとともに返す。
    - セッション ID (セッションのユニークな識別子)
    - セッションのステータス：
      - *active*
      - *inactive*
    - *active remotely or inactive* – セッションのステータスの明確な判別が不可能。現在のインスタンスと同じディレクトリを使用している、Historical Server の別のインスタンスで現在アクティブなセッションは、このカテゴリに入る (Historical Server の複数のインスタンスの実行と制御ファイルの共有については、「[Historical Server の複数インスタンスの設定 \(23 ページ\)](#)」を参照)。セッションを実行していた Historical Server インスタンスの異常終了によりすでに終了したセッションで、終了時刻をまだ過ぎていなかったものも、このカテゴリに入る。
  - レコーディング・セッションを開始したユーザ名
  - モニタしている Adaptive Server の名前
  - 使用している Monitor Server の名前
  - レコーディング・セッションの開始日時
  - レコーディング・セッションの終了日時
  - 記録データを保持するディレクトリ
  - 使用されるサンプル・インターバル
- この値が 0 の場合、`summarization_level` が `entire` であるプレイベックによってセッションが作成されたことを示します (セッション全体が 1 つのサンプルで表されるので、サンプル・インターバルは存在しません)。
- 使用するエラー・オプション (致命的でないエラーについて `continue` または `halt`)
  - `views` は、ビューごとに 1 つのローを以下のカラムとともに作成する。
    - セッション ID
    - ビュー名

- **data\_items** は、ビューで定義されたデータ項目ごとに 1 つのローを以下のカラムとともに作成する。
  - セッション ID
  - ビュー名
  - データ項目
  - データ項目の統計タイプ
  - データ項目に対して定義されているアラームの数
  - データ項目に対して定義されているフィルタの数
  - 次のキーワードのいずれか。
    - **recorded** – データがオリジナルのレコーディング・セッションから得られたものか、オリジナル・セッションから保存されたものであることを示す。
    - **summarized** – データがオリジナル・セッションとは異なるサンプル・インターバルで要約されたことを示す。
    - **estimated** – データが、要約化の際に正確に計算されたのではなく、推定されたものであることを示す。
- **alarms** は、ビューのデータ項目に対して定義されたアラームごとに 1 つのローを以下のカラムとともに作成する。
  - セッション ID
  - ビュー名
  - データ項目
  - データ項目の統計タイプ
  - アラーム・アクション (log または execute)
  - ログ・ファイル名または実行するファイルの名前
  - アラーム値
- **filters** は、ビューのデータ項目に対するフィルタごとに 1 つのローを以下のカラムとともに作成する。
  - セッション ID
  - ビュー名
  - データ項目
  - 統計タイプ
  - フィルタ・タイプ (eq、 neq、 range、 または top)
  - 指定した値 (1 つの文字列として返される)

- `summarization_data_items` は、ロー以外のプレイバックに対してビューから要求できる、データ項目と統計タイプの組み合わせごとに1つのローを以下のカラムとともに作成する。
  - セッション ID
  - ビュー名
  - データ項目
  - 統計タイプ
  - 次のキーワードのいずれか
    - `recorded` – プレイバック時に、このデータ項目用のデータがオリジナルのレコーディング・セッションから得られたデータ、またはそのオリジナル・セッションから保持されていたデータとなる。
    - `summarized` – プレイバック時に、このデータ項目用のデータが概要フォームで利用可能になる。
    - `estimated` – プレイバック時に、このデータ項目用のデータが推定フォームのみで利用可能になる。

## hs\_playback\_sample

### 説明

プレイバック `target` が `client` である場合にサンプルをプレイバックします。`target` は、`hs_create_playback_session` コマンドで定義します。

### 構文

`hs_playback_sample [step [, retry_count] ]`

### パラメータ

`step`  
クライアントに送信するサンプルを、現在のプレイバック・セッションに対して最後に実行された `hs_playback_sample` コマンドで送信されたサンプルに相対的に指定します。有効な値は 0 または正の数です。

デフォルトは +1 で、これは 1 つ後のサンプルを送信します。はじめに、+1 の `step` は、最初のサンプルを返します。0 にすると、最も新しく送信されたサンプルが再送信されます。+2 の `step` では、+1 の `step` で送信されたサンプルの次に新しいサンプルが送信されます。最新のサンプルよりも新しいサンプルを表示させようとすると、エラー・メッセージが返されます。

`hs_initiate_playback` コマンドはプレイバック・セッションを初期化し、デフォルト (+1) でプレイバック・セッションの最初のサンプルが送信されるようにします。

サンプルの構成要素の定義は、`hs_create_playback_session` コマンドの `summarization_interval` パラメータと `missing_data_option` パラメータの影響を受けます。

`actual`、`entire`、またはユーザ指定のインターバルの `summarization_interval` パラメータを使用してプレイバック・セッションを作成した場合、また、プレイバック・セッションのいずれかのビューが、統計タイプ “Value for Session” または “Rate for Session” のデータ項目を含む場合、`step` に許可される唯一の値はデフォルトの +1 です。

*retry\_count*

プレイバック時にまだ記録中であるレコーディング・セッションの履歴データ・ファイルからデータを読み込むために、1 秒間に Historical Server がリトライする回数を指定します。

デフォルトは 0 です。つまり、履歴データ・ファイルの読み込み中にプレイバックがファイルの終端の状態になった場合、リトライは行われず、エラーが返されます。

例

- 1 この例では、次のサンプルを取り出します。

```
hs_playback_sample
```

- 2 この例では、1 つのサンプルをスキップし、次のサンプルをプレイします。レコーディングの継続時において、ファイルの終端に達した場合、Historical Server は他のサンプルを取り出すことを 30 回目までリトライします。

```
hs_playback_sample 2, 30
```

使用法

返されるデータのフォーマット

`hs_playback_sample` コマンドは、テーブルの形式でデータを返します。テーブルは、プレイバック・ビューごとに返されます。これらのテーブルは、ビューを作成した `hs_create_playback_view` コマンドを送信したときと同じ順序で並びます。

各テーブルのカラムは、対応するプレイバック・ビューのデータ項目と対応します。これらのカラムは、ユーザがプレイバック・ビューを定義するときにデータ項目をリストした順序で並びます。各カラムのデータ型を確認するには、[表 B-3 \(145 ページ\)](#) を参照してください。

各ローは、ビューにあるキー・データ項目の異なる組み合わせを表しています。キー・データ項目がビューにない場合、テーブルは、サーバレベルのデータを反映する单一のローを返します。

報告するアクティビティがないときは、一部のビューはゼロのついたローを返し、他のビューはこのローを省略します。ゼロ個の値を持つローがビューに表示されるかどうかを制御する規則は、次のとおりです。

- サーバ・レベルのデータ項目を持つビューは、報告するアクティビティがない場合でも、常にローを返す。
- キー・データ項目 [Process ID]、[Object ID]、または [Procedure ID] を含むビューは、報告するアクティビティがないときにはそのローを省略する。プロセス ID を返すタイミングについては、「[\[Process ID\] を含むビュー](#)」(77 ページ) を参照。
- 前の項目にリストされているキー以外のキーを含むビューは、アクティビティがない場合でもローを返す。

プレイバックによってデータが要約されている場合、入力セッションのいずれかに同じキーの組み合わせを含むサンプルがあると、キーの組み合わせについてのローが返されます。

整数データがオーバフローしたときは、Historical Server はそのデータ項目で有効な最大の整数を返し、クライアントに情報メッセージを返します。

#### [Process ID] を含むビュー

サーバ・プロセスが終了すると、Adaptive Server はそのプロセス ID を新しいプロセスのために再使用することがあります。したがって、[Process ID] データ項目では、プロセスをユニークに識別できるとは限りません。ただし、[Kernel Process ID] データ項目を使用すると、プロセスをユニークに識別できます。

Process ID を含むビューは、次のようにローを返します。

- レコーディング・セッション・ビュー(およびロー・プレイバック・ビュー)は、サンプル間隔の最後に存在したプロセスを表すプロセス ID のローだけを返す。サンプル間隔の途中でサーバ・プロセスが終了した場合、そのプロセス ID のローは返さない。
- 概要プレイバック・ビューは、サーバ・プロセスがプレイバック・サンプル中に存在し続けることを要求しない。概要ビューは、入力ビューのいずれかに含まれているサーバ・プロセスすべてのローを返す。ただし、プロセス ID がユニークであるという保証はないため、キーの一意性を保証するには、[Process ID] を使用するすべての概要ビューで [Kernel Process ID] も使用する必要がある。このデータ項目がないと、誤って 2 つの異なるプロセスのデータが結合される可能性がある。

### プレイバック・セッションでのデータのギャップ

`hs_create_playback_session` コマンドの *start\_time* パラメータと *end\_time* パラメータは、プレイバック・セッションが適用される時間を定義します。`hs_create_playback_session` コマンドの *target* パラメータが *client* の場合、どの入力セッションにも利用できるデータがないプレイバック・セッションの開始時刻と終了時刻の間に空白の時間が生じることがあります。

---

**注意** *target* が *file* の場合、指定した入力セッションでギャップは許可されません。したがって、プレイバック・セッションにギャップが存在することはありません。

---

利用することのできるデータにギャップが存在する可能性があるのは、次の場所です。

- プレイバック・セッションに指定された入力セッションが連続していない場合。たとえば、一連の入力セッションを月曜日～金曜日の午前 9 時～午後 5 時に適用している場合、夕方の午後 5 時から翌朝の午前 9 時までの間、データのギャップが存在することになります。さらに大きなギャップが、金曜日の夕方から月曜日の朝にかけて存在します。
- プレイバック・セッションによって要約が行われ、入力セッションからデータを取得できない開始時刻または終了時刻がある場合。

プレイバックのターゲットが *client* の場合、Historical Server は、以下のように `hs_create_playback_session` コマンドの *missing\_data\_option* パラメータの値に従ってギャップを処理します。

- *skip* (デフォルト) – ある期間のデータが存在しない場合、`hs_playback_sample` コマンドはデータが存在しないサンプルを返すのではなく、データが存在する次の期間に直接移行します。
- *show* – `hs_playback_sample` コマンドは、利用できるデータがない時間についてもサンプルを返します。データは返されません (返される各データ・テーブルの長さはゼロです)。

この場合にクライアントがタイムスタンプを受け取るには、データ項目 *[Timestamp]* または *[Timestamp Datim]* だけを含むプレイバック・ビューが必要です。ビューがそれら以外のデータ項目を含む場合、ゼロ個のローを返します。

プレイバック・セッションの概要レベルは、プレイバック中に返された空のサンプル数に影響を与えます。

- *summarization\_level* が **raw** または **actual** の場合、ギャップ全体が 1 つのサンプルで表されます。たとえば、午後 5 時と午前 9 時の間のデータのギャップは 1 つのサンプルによって表されます。
- *summarization\_level* が 30 分などの時間間隔 ("0 00:30:00" のように指定) ある場合、ギャップは各時間間隔にサンプルとともに表されます。たとえば、時間間隔が 30 分を指定し、午後 5 時と午前 9 時の間のデータのギャップは 32 個のサンプルによって表されます。
- *summarization\_level* が **entire** の場合、ギャップは無視されます。

## hs\_shutdown

### 説明

Historical Server をシャットダウンします。

### 構文

`hs_shutdown [wait_option]`

### パラメータ

*wait\_option*  
終了をすぐに行うか、アクティブな接続がクローズするのを待つかを指定します。有効な値は次のとおりです。

- **no\_wait** – Historical Server をすぐにシャットダウンし、( プレイバックを実行中のものを含む ) すべての接続とアクティブなレコーディング・セッションを終了させます。新しいセッションを作成していたアクティブなレコーディング・セッションとプレイバック・セッションは、制御された方法でシャットダウンします。これはコントロール・ファイルや履歴データ・ファイルの破壊を防ぎます。
- **wait** (デフォルト) – ( プレイバックを実行中のものを含む ) すべてのアクティブなレコーディング・セッションが終了し、その他のすべてのアクティブな接続がクローズするまでシャットダウンを遅らせます。この間、新規の接続は許可されません。

### 例

この例では、Historical Server をすぐにシャットダウンし、すべての接続とアクティブなレコーディング・セッションを終了させます。

```
hs_shutdown no_wait
```

## 使用法

- no\_wait オプションを指定しないで `hs_shutdown` コマンドを使用し、その要求をキャンセルする場合、ユーザは割り込み指示を Historical Server に送信してシャットダウン要求をキャンセルできます。DB-Library アプリケーションは `dbcancel()` 関数を、CT-Library アプリケーションは `ct_cancel()` 関数をそれぞれ呼び出して、こうした割り込みを発生させます。ユーザが割り込みシグナルを送信すると、`isql` コマンドによって、これらの呼び出しのいずれかが生成されます。割り込みシグナルは、通常、[Ctrl] キーを押しながら [C] キーを押すことで送信されます。
- 以下の 4 つの信号のシグナルのいずれかを送信することで、いつでも Historical Server の即時シャットダウンを要求することができます。
  - SIGQUIT (quit)
  - SIGINT (interrupt)
  - SIGABORT (abort)
  - SIGTERM (terminate)

これらのシグナルのいずれかをどの時点で受信した場合でも (`hs_shutdown` コマンドの完了を待っている間も含む)、`hs_shutdown no_wait` コマンドを発行したことと同じになります。

---

**警告！** Sybase では、`kill` シグナル (SIGKILL) を使用して Historical Server をシャットダウンすることは避けるよう強くおすすめします。`kill` シグナルは、制御されたクリーンアップを許容しないため、結果的にデータの損失につながることがあります。

---

- シャットダウン中にすべてのアクティブなセッションが終了すると、Historical Server はログ・ファイルにそのセッションについての情報を書き込みます。アクティブなセッションは、新しいレコーディング・セッションを作成するプロセスにおけるレコーディング・セッションやプレイバック・セッションです。
- Historical Server の起動コマンドにスーパーユーザが指定された場合、Historical Server をシャットダウンできる唯一のユーザが、そのスーパーユーザです。スーパーユーザが指定されなかった場合、すべてのユーザが Historical Server を停止できます。
- Historical Server の停止の詳細については、「[第 3 章 Historical Server の起動と停止](#)」を参照してください。

## hs\_status

説明

ステータス情報を取得します。

構文

`hs_status option`

パラメータ

次のいずれかを指定します。

- `directory` – Historical Server のホーム・ディレクトリの名前を表示します。ホーム・ディレクトリは、Historical Server 起動コマンドの `-D` パラメータで指定されます。
- `superuser` – Historical Server のスーパーユーザの名前を表示します。スーパーユーザの名前は、Historical Server 起動コマンドの `-U` パラメータで指定されます（このパラメータを省略した場合は `NULL` です）。
- `interfaces` – Historical Server が使用中の `interfaces` ファイルまたは `sql.ini` ファイルのパス名を表示します。
- `max_connections` – 許可されている同時クライアント接続の最大数を表示します。
- `logfile` – Historical Server ログ・ファイルの名前を表示します。
- `version` – Monitor Historical Server のバージョン文字列と版権情報を表示します。
- `activity` – Historical Server の現在のアクティビティ・レベルについての以下の情報を表示します。

接続の数	現在の接続を含む Historical Server への接続
アクティブなレコードイング・セッションの数	<code>hs_create_recording_session</code> で作成され、 <code>hs_initiate_recording</code> で開始されたセッション。まだ開始時刻になっていないセッションも含む。
開始していないレコードイング・セッション	<code>hs_create_recording_session</code> で作成されたが、対応する <code>hs_initiate_recording</code> が実行されなかったセッション。
アクティブなプレイバック・セッションの数	<code>hs_create_playback_session</code> で作成され、 <code>hs_initiate_playback</code> で開始されたセッション。
開始していないプレイバック・セッションの数	<code>hs_create_playback_session</code> で作成されたが、対応する <code>hs_initiate_playback</code> が実行されなかったセッション。

アクティビティ情報が関与するのは、Historical Server を停止するときです。`hs_shutdown` コマンドの `no_wait` オプションを明示的に要求しないかぎり、Historical Server は他の接続やアクティブなセッションが存在する間はシャットダウンしないためです。

## hs\_terminate\_playback

---

例	この例では Historical Server のホーム・ディレクトリの名前を表示します。
使用法	<pre>hs_status directory</pre> <p><b>hs_status</b> コマンドは、<b>max_connections</b> フィールドとすべての <b>activity</b> フィールドを <b>integer</b> データ型として返します。すべての他のフィールドは文字列として返されます。</p>

## **hs\_terminate\_playback**

説明	プレイバック・セッションの定義と実際のプレイバックを終了します。
構文	<pre>hs_terminate_playback</pre>
使用法	<ul style="list-style-type: none"><li>このコマンドが有効になるのは、<b>hs_create_playback_session</b> コマンドが正常に発行された後だけです。</li><li><b>hs_create_playback_session</b> コマンドを正常に発行したら、<b>hs_terminate_playback</b> を発行してから追加のプレイバック・セッションやレコーディング・セッションの定義を開始します。</li><li>レコーディング・セッションの定義とは異なり、プレイバック・セッションの定義は、Historical Server コントロール・ファイルに格納されません。</li></ul>

## **hs\_terminate\_recording**

説明	レコーディング・セッションの定義を終了し、スケジュール設定されたレコーディング・セッションの開始をキャンセルし、また進行中のレコーディング・セッションを終了します。
構文	<pre>hs_terminate_recording session_id [,delete_option]</pre>
パラメータ	<p><b>session_id</b> 終了するセッションを指定します。<b>hs_create_recording_session</b> でセッションを定義するときに Historical Server はセッション ID を割り当てます。セッション ID を表示させるには <b>hs_list</b> コマンドを使用します。</p> <p><b>delete_option</b> レコーディングが進行中の場合、レコーディング・セッションに関連するファイルを削除するかどうかを指定します。</p> <ul style="list-style-type: none"><li><b>delete</b> – ファイルを削除します。</li><li><b>retain</b> (デフォルト) – ファイルを削除しません。開始していないレコーディング・セッションをキャンセルする場合、このパラメータは無視されます。</li></ul>

## 例

この例では、識別子 5 を持つアクティブなレコーディング・セッションを終了し、セッション 5 に関連するすべてのファイルを削除します。

```
hs_terminate_recording 5,delete
```

## 使用法

- ユーザがレコーディングを開始した場合、そのユーザが、セッションの所有者になります。
- レコーディング・セッションを終了するには、セッションの所有者が Historical Server のスーパーユーザであることが必要です。
- `hs_terminate_recording_session` コマンドは、現在の接続で作成され、まだ開始されていないレコーディング・セッションの定義をキャンセルできます。これにより、ユーザは現在の接続を利用して、他のレコーディング・セッションを定義したり、プレイバック・セッションを定義したりすることができます。
- 非アクティブのレコーディング・セッションを終了しようとした場合、このコマンドはエラーを返します。

hs\_terminate\_recording

---

トピック名	ページ
Historical Server データ・ファイルの概要	85
制御ファイル	86
データ・ファイル	91
エラー・メッセージ・ファイル	92
スクリプト・ファイル	92
バルク・コピーの例	94
Cut ユーティリティの例	96
Historical Server でモニタリング・データのデータベースへの出力を可能にする	97

## Historical Server データ・ファイルの概要

この項では、次の内容について説明します。

- [Historical Server ファイルの説明](#)
- [ファイルのパーミッション](#)
- [一般的なファイル・フォーマット](#)

## Historical Server ファイルの説明

Historical Server には次のファイルがあります。

- [制御ファイル](#) – 同じ Historical Server のすべてのレコーディング・セッションは、同じ制御ファイルを使用します。Historical Server の複数のインスタンスと制御ファイルについては、「[Historical Server の複数インスタンスの設定](#)」(23 ページ) を参照してください。
- [エラー・メッセージ・ファイル](#) – 各レコーディング・セッションに対して個別のエラー・メッセージ・ファイルが作成されます。セッションでエラーが記録されなかった場合、ファイルは空です。
- [データ・ファイル](#) – レコーディング・セッションの各ビューは、データ・ファイルで表されます。データ・ファイルには、セッション中にビューに対して収集されたすべてのサンプルが収められます。

- スクリプト・ファイル – `hs_create_recording_session` コマンドを使用してレコーディング・セッションを定義した場合、`script_type` パラメータを使用して SQL スクリプト・ファイルの作成を指定できます。スクリプト・ファイルでは、レコーディング・セッションの各ビューの Adaptive Server テーブルを定義します。

バルク・コピー (`bcp`) ユーティリティを使用して、既存の Adaptive Server テーブルに移植できます。UNIX ユーザの場合、まず `cut` ユーティリティを起動してファイルから必要なカラムを抜き出すと、フィールドのサブセットを別の Adaptive Server テーブルに置くことができます。

## ファイルのパーミッション

特定のレコーディング・セッションのすべての履歴モニタリング・データ・ファイルは、同じディレクトリに格納されます。デフォルトでは、このディレクトリは、Historical Server 起動コマンドの `-D` パラメータで指定したディレクトリです。セッションのデフォルトは、セッションの作成時に上書きできます。

Historical Server を起動するユーザが、その Historical Server で作成されるファイルの所有者です。所有者には、履歴モニタリング・ファイルに対する読み込みおよび書き込みパーミッションが与えられます。履歴データ・ファイルまたはエラー・メッセージ・ファイルに含まれるデータがプライベートではなくパブリックの場合、すべてのユーザとグループに読み込みパーミッションが与えられます。

## 一般的なファイル・フォーマット

すべてのファイルで、レコード終端マーカとして改行文字を使用します。各ファイルのフィールドは可変長で、カンマを使用して区切れます。データはすべて、Sybase バルク・コピー (`bcp`) ユーティリティと互換性のある ASCII フォーマットで格納されます。

## 制御ファイル

`hs.ctl` 制御ファイルは、Historical Server のホーム・ディレクトリにあります。制御ファイルには、Historical Server が認識している過去と現在のすべてのレコーディング・セッションに関する情報が収められます。Historical Server での制御ファイルの機能の詳細については、「[Historical Server の制御ファイルとホーム・ディレクトリ](#) (9 ページ)」を参照してください。

Historical Server の `hs_list` コマンドを使用して、Historical Server 制御ファイルの情報にアクセスできます。制御ファイルを編集しないでください。ファイルが壊れることがあります。

どのようなエディタであっても、エディタで制御ファイルを開き、保存することは行わないでください。Historical Server を Windows で実行する場合には特に注意が必要です。Historical Server によって作成される他のファイルと異なり、制御ファイルは Windows の標準フォーマットのテキスト・ファイルではありません。制御ファイルのテキスト行を終了するには、改行文字だけを使用する必要があります。通常の復帰／改行の組み合わせは使用しません。編集プログラムを使用すると、テキストに不要な復帰／改行文字が埋め込まれ、ファイルが壊れてしまう可能性があります。

制御ファイルは、次のタイプのレコードで構成されています。

- ヘッダ・レコード
- セッション制御レコード
- データ項目制御レコード
- アラーム制御レコード
- ビュー制御レコード
- フィルタ制御レコード

## ヘッダ・レコード

制御ファイルの最初のレコードは、次の 6 つの整数フィールドからなるヘッダ・レコードです。

- 「更新中」 フラグ
- ファイル内で更新されている情報のバイト・オフセット
- Historical Server によって次のレコーディング・セッションに割り当てられるユニーク・セッション ID
- ファイルの破損の検出とリカバリに使用するユニーク・セッション ID のコピー
- 制御ファイルに制御情報が最後に書き込まれたセッションのセッション ID
- セッション・エントリが参照する 1 つ以上のデータ・ファイルまたはエラー・メッセージ・ファイルが削除されたために無効になった、制御ファイル内のセッション・エントリの数

このヘッダ・レコードの後ろに、Historical Server が認識する過去と現在すべてのレコーディング・セッションを説明するさまざまなタイプの多数のレコードが続きます。これらのレコードは、制御ファイルの階層順序に従って表示されます。レコード・タイプの階層は上から順に次のとおりです。

- セッション制御
- ビュー制御

- データ項目制御
- アラーム制御およびフィルタ制御

たとえば、それぞれが 1 つのアラームと 1 つのフィルタを含む 2 つのデータ項目からなる 2 つのビューで構成されるセッションは、次の制御レコード階層で表され、次の順序で格納されます。

```
session 1
    view 1
        data item 1
            alarm 1
            filter 1
        data item 2
            alarm 2
            filter 2
    view 2
        data item 3
            alarm 3
            filter 3
        data item 4
            alarm 4
            filter 4
```

次の項からは、各制御レコード・タイプの内容について説明します。

## セッション制御レコード

セッション制御レコードは次の情報で構成されています。

- レコード識別子 (“session” という単語)
- セッションのステータス。ステータスは、アクティブ、非アクティブ、無効のいずれかです。無効ステータスは、セッションに属するデータ・ファイルまたはエラー・メッセージ・ファイルが削除されたか、起動時間に達する前にセッションが終了された場合に適用されます。
- このレコードの始まりから数えた、制御ファイルの次のセッション・レコードまでのバイト・オフセット
- セッション ID。この制御ファイルに含まれるすべてのセッションの中から特定のセッションをユニークに識別する番号。セッション ID は、昇順に順次割り当てられる整数です。
- レコーディングを要求したクライアントのユーザ名

- レコーディング・セッションの開始日付と時刻

制御ファイルの開始時刻と終了時刻には、レコーディング・セッションが実際に開始および終了したできるだけ正確な時刻が反映されます。これらの実際の時間は、コマンドで指定した *start\_time* パラメータおよび *end\_time* パラメータの値と少し異なることがあります。たとえば、*start\_time* が午後1時としてスケジュールされていても、システムがビジー状態のために、セッションが実際には午後1時2分に開始することがあります。

- レコーディング・セッションの終了日付と時刻
- モニタしている Adaptive Server の名前
- レコーディング・セッションに参加している Monitor Server の名前
- セッションのモニタリング・ファイルが格納されるディレクトリのパス名
- セッションのエラー・メッセージ・ファイルの名前
- レコーディング中に使用するサンプル・インターバル(秒)

この値が 0 の場合、*summarization\_level* が *entire* であるプレイバックによってセッションが作成されたことを示します(セッション全体が 1 つのサンプルで表されるので、サンプル・インターバルは存在しません)。

- 有効な保護レベル (public/private)
- エラー・オプション (continue/halt。レコーディング中に致命的ではないエラーが発生した場合)
- 作成されたスクリプト・ファイルのタイプ (sybase\_script/no\_script)
- レコーディング・セッションを作成したクライアントの暗号化されたパスワード
- セッションがプレイバックの結果作成された場合、キーワード *summary*

ファイルには、レコーディング・セッションごとに 1 つのセッション制御レコードがあります。そのすぐ後ろに、セッションの最初のビュー制御レコードが続きます。

## ビュー制御レコード

ビュー制御レコードは次の情報で構成されています。

- レコード識別子 (“view” という単語)
- ビューネ名
- ビューデータ・ファイルの名前

レコーディング・セッションで定義されたビューごとに 1 つのビュー制御レコードが存在します。各制御レコードの後ろに、ビューのデータ項目を識別する 1 つ以上のデータ項目制御レコードが続きます。アラーム制御レコードとフィルタ制御レコードは、データ項目制御レコードの間に適宜分散して挿入されます。

## データ項目制御レコード

データ項目制御レコードは次の情報で構成されています。

- レコード識別子 (“dataitem” という単語)
- データ項目名
- 統計タイプ
- プレイバックから作成されたセッションの場合、キーワード **estimated** は、他のセッションのデータ項目からの推定によって値を取得したデータ項目を示します。

現在のビューの定義を構成するデータ項目ごとに 1 つのデータ項目制御レコードが存在します。アラームかフィルタ、またはその両方がデータ項目に適用される場合、そのデータ項目の制御レコードの後ろにこれらの制御レコードが続きます。データ項目のすべてのアラーム制御レコードが、同じデータ項目のフィルタ制御レコードの前に格納されます。

## アラーム制御レコード

アラーム制御レコードは次の情報で構成されています。

- レコード識別子 (“alarm” という単語)
- アラーム・アクション
- アラーム・アクション・データ (ファイル名。その後ろにパラメータのリストが続く場合があります)
- アラームのデータ型
- アラーム値

現在のデータ項目に定義されているアラームごとに 1 つのアラーム制御レコードが存在します。データ項目のすべてのアラーム制御レコードは、データ項目の最初のフィルタ制御レコードの前にまとめられます。

## フィルタ制御レコード

フィルタ制御レコードには次の情報が含まれます。

- レコード識別子 (“filter” という単語)
- フィルタ・タイプ
- フィルタのデータ型
- 値の仕様 (`hs_create_filter` コマンドで使用されているフォーマットと同じく、カンマで区切られた 1 つ以上のフィールド)

現在のデータ項目に適用されたフィルタごとに 1 つのフィルタ制御レコードが存在します。

## データ・ファイル

データ・ファイルには、1 つのセッション中に記録された 1 つのビューのデータが収められます。ファイル名は、`dsessionid_viewnumber_viewname` です。

- `sessionid` は、セッションのユニークな識別子です。
- `viewnumber` は、セッションでビューが定義された順序を示す 1 以上の数字です。
- `viewname` は、ビューのユーザ指定名です。現在のプラットフォームのファイル命名規則に従って、必要に応じてトランケートされます。Windows プラットフォームでは、ファイル名は 8 文字に制限されます。

ファイルには、取り出した各サンプルで発生したデータ項目キー・フィールド値のユニークな組み合せごとに 1 つのレコードが存在します。

各レコードには次の情報が収められます。

- サンプルの日付と時刻  
このタイムスタンプは、モニタした Adaptive Server で有効なローカル時間に基づきます。
- サンプリングしたすべてのデータ項目の値

特定のサンプル・インターバル中にこのビューに対してデータが返されない場合、履歴データ・ファイルにはプレースホルダ・レコードが書き込まれます。このレコードでは、サンプルのタイムスタンプの後ろに、カンマで区切られた空の値のリストが続きます。つまり、タイムスタンプ・フィールドの後ろにビューの残りのデータ項目を区切るために必要な数のカンマが続きますが、これらのデータ項目の値は格納されません。このフォーマットは、データ項目のコピー先のカラムが `null` を受け入れる場合は、パルク・コピー・ユーティリティとも互換性があります。

## エラー・メッセージ・ファイル

セッションのサンプルの記録中に受け取ったエラー・メッセージは個別のエラー・メッセージ・ファイルに格納されます。このファイルの名前は *eSessionId* です。

- *e* は定数です。
- *SessionId* は、セッションのユニークな識別子です。

このファイルの各レコードには、次のフィールドがあります。

- エラー・メッセージが発生したサンプルの日付と時刻。このタイムスタンプは、モニタした Adaptive Server で有効なローカル時間に基づきます。エラーによって、Historical Server がサンプルのタイムスタンプを取り出せない場合、タイムスタンプ・フィールドは空になることがあります。
- エラー番号
- エラー重大度
- エラーのステータス
- エラーのソース
  - 1 – 不定
  - 2 – Historical Server
  - 3 – CT-Library
  - 4 – Adaptive Server
  - 5 – Monitor Server
- エラー・メッセージ・テキスト

エラー・メッセージ・フィールドの詳細については、『リファレンス・マニュアル』を参照してください。

## スクリプト・ファイル

`hs_create_recording_session` コマンドの *script\_type* パラメータの値が `sybase_script` の場合、セッションのスクリプト・ファイルが作成されます。このスクリプト・ファイルの名前は *ssessionid* です。*sessionid* は、セッションのユニークな識別子です。

## スクリプト・ファイル・テーブル名

スクリプト・ファイルは `create table` コマンドのテキストで構成されています。コマンドは、セッションに定義されたビューごとに1つ使用します。各テーブルの名前は、`sSessionId_viewnumber_viewname` です。

- *s* は定数です。
- *SessionId* は、セッションのユニークな識別子です。
- *viewnumber* は、セッションでビューが定義された順序を示す 1 以上の数字です。
- *viewname* は、ビューのユーザ指定名です。Adaptive Server 識別子の長さの制限に従って、必要に応じてトランケートされます。

## スクリプト・ファイルのテーブル・カラム名

各テーブルで最初に定義するカラムは、サンプルのタイムスタンプ・フィールドです。名前は常に “Timestamp” で、データ型は `datetime` です。

残りのテーブル・カラムの名前は、それぞれが表すデータ項目から導出されます。各カラム名は `dataitemname_stattype` です。

- *dataitemname* は、データ項目の名前の圧縮形です。
- *stattype* は、データ項目の統計タイプの省略形です。

各カラムには、データ項目が表す情報のタイプに適したデータ型が割り当てられます。サンプル・タイムスタンプ・カラムを除くすべてのカラムには、`null` 値を格納できます。

## スクリプト・ファイル・コマンドの受け渡し

スクリプト・ファイルの `create table` コマンドは、`isql` ユーティリティの `:r` コマンドを使用して Adaptive Server に渡すことができます。`:r` コマンドを使用してテーブルを定義したら、バルク・コピー・ユーティリティを使用して、ビューのデータ・ファイルをテーブルにインポートできます。バルク・コピーの例については、[「バルク・コピーの例」\(94 ページ\)](#) を参照してください。

## スクリプトの使用例

次の `isql` コマンドを使用して、記録したモニタリング・データを格納する Adaptive Server に接続します。

```
isql -Uusername -Ppassword -Sserver
```

`username` には、その Adaptive Server のログイン・アカウント名、`password` には、そのログイン・アカウントのパスワード、`server` には、Adaptive Server の名前を指定します。

次の `isql` コマンドを実行します。

```
use database
go
```

`database` には、新しいテーブルを格納するデータベースの名前を指定します。

```
:r scriptfile
go
```

`scriptfile` には、Historical Server によって作成されたスクリプト・ファイルのパス名を指定します。

この結果、レコーディング・セッションに対して定義されたビューごとに、Adaptive Server に 1 つのテーブルが存在します。

## バルク・コピーの例

この項では、履歴データ・ファイルをバルク・コピー (`bcp`) ユーティリティに渡して、Adaptive Server テーブルに移植する方法を説明します。

`bcp` を使用して、記録されたデータを Adaptive Server テーブルに格納するには、まずセッションに定義されたビューごとにテーブルを作成します。テーブルでは、ビューを構成するデータ項目ごとに個別のカラムを作成します。各テーブルの最初のカラムには、サンプルの日付と時刻を格納し、`datetime` データ型にする必要があります。

Historical Server が記録している間、特定のサンプル・インターバル中にこのビューに対してデータが返されない場合、ビューのデータ・ファイルにはプレースホルダ・レコードが書き込まれます。このレコードには、サンプルのタイムスタンプの後ろに、カンマで区切られた空の値のリストが続きます。このため、最初のカラム以外のカラムは、`null` を受け入れるように定義してください。

Historical Server でスクリプト・ファイルを作成することにより、テーブル作成プロセスを簡略化できます。`hs_create_recording_session` コマンドの `script_type` パラメータを使用します。

Adaptive Server にテーブルを作成したら、**bcp** ユーティリティを使用して、ビューのデータ・ファイルからそれぞれに対応するテーブルにデータをインポートできます。データを Adaptive Server テーブルにインポートするデータ・ファイルごとに、次のコマンドを 1 回実行します。

```
bcp [[database_name.]owner.]table_name in view_data_file -c [-e errfile] [-U username] [-P password] [-S sqlserver] -t,
```

各パラメータの意味は、次のとおりです。

- *database\_name* は、インポートしたデータを保持するテーブルを格納するデータベースの名前。
- *owner* は、テーブルの所有者の名前(多くの場合、テーブルを保存するデータベースのユーザ名)。
- *table\_name* は、ビューのデータ・ファイルのデータをインポートするデータベース・テーブルの名前。
- *view\_data\_file* は、ビューのデータ・ファイルの名前。
- *errfile* は、データベース・テーブルに転送できなかったローが格納されるエラー・ファイルの名前。
- *username* は、Adaptive Server へのログイン名。
- *password* は、*username* のパスワード。
- *sqlserver* は、Adaptive Server の名前。
- *-t* には、カラム・ターミネータをカンマに指定。

*-c* オプションを使用した場合のデフォルトのロー・ターミネータは、データ・ファイルで使用される *¥n* (改行) 文字です。したがって、*-r* オプションを指定する必要はありません。レコード・ターミネータ・オプション (*-r*) を指定する場合は、*-r¥¥n* のように、円記号 (¥) を 2 つ使用して、円記号が特別な意味を持たないようにする必要があります。

**bcp** の詳細については、『ユーティリティ・ガイド』を参照してください。

## 例

SMS1100 Monitor Server でモニタされている Adaptive Server の各デバイスについて、1 秒に 1 回、読み込み、書き込み、I/O の総数を記録するとします。次の手順に従って、データを記録し、データを受け入れるテーブルを作成し、**bcp** ユーティリティを使用してテーブルにデータをインポートできます。この例では、Historical Server によって作成されるビューのデータ・ファイルは *d1\_1\_device\_io* です。

1 isql を使用して Historical Server にログインします。

2 次のコマンドを使用して、データを記録します。

```
hs_create_recording_session SMS1100, 1
go
hs_create_view device_io,
"device name", "value for sample",
"device reads", "value for sample",
"device writes", "value for sample",
"device i/o", "value for sample"
go
hs_initiate_recording
go
```

3 isql を使用して Adaptive Server にログインします。レコーディング・セッションに含まれているビューは 1 つだけ (“device\_io”) なので、pubs2 データベースにテーブルを 1 つだけ作成し、次の Transact-SQL コマンドを使用して、そのビューのデータを格納します。

```
use pubs2
go
create table device_io
(
sample_time      datetime      not null,
device_name      varchar (255)  null,
device_reads_val int          null,
device_writes_val int          null,
device_io_val    int          null,
)
go
```

4 レコーディング・セッションが完了したら、bcp を使用して、ビューのデータ・ファイルからデータベース・テーブルにデータをインポートできます。次のコマンドを使用します。

```
bcp pubs2..device_io in d1_1_device_io -c
-e d1_1_device_io.err -Username
-Password -Sserver -t
```

## Cut ユーティリティの例

UNIX マシンの場合、まず cut ユーティリティを起動し、ファイルから必要なカラムを抜き出して保存すると、データ・ファイルのフィールドのサブセットを別の Adaptive Server テーブルに置くことができます。次のコマンドを使用します。

```
cut -flist -d, view_data_file > output_file_name
```

各パラメータの意味は、次のとおりです。

- *list* は、オリジナルのビューのデータ・ファイルからカットし、保存するフィールド番号をカンマで区切ったリスト。最初のフィールドのフィールド番号は 1。
- *view\_data\_file* は、すべてのレコーディング・データが格納されるファイルの名前。
- *output\_file\_name* は、抜き出したデータを格納するファイルの名前。

*cut* の使用後、*bcp* ユーティリティを使用して、*output\_file\_name* のデータを Adaptive Server テーブルにインポートできます。Historical Server から、次のような行が *view\_data\_file* に返されます。

```
1995/3/25 10:32:39,master,0,0,0
1995/3/25 10:32:39,sysprocsdev,0,0,0
1995/3/25 10:32:39,ANewDBDevice,0,0,0
1995/3/25 10:32:39,ATestDevice,0,0,0
1995/3/25 10:32:40,master,0,0,0
1995/3/25 10:32:40,sysprocsdev,0,0,0
1995/3/25 10:32:40,ANewDBDevice,0,0,0
1995/3/25 10:32:40,ATestDevice,0,0,0
```

サンプル中、各デバイスの I/O の総数だけを取り出したいとします。次のコマンドを実行して、必要なデータを取り出すことができます。

```
cut -f1,2,5 -d, d1_1_device_io > d1_1_device.io.new
```

フィールド 1 はサンプルの日付と時刻、2 はデバイス名を含む第 2 フィールド、5 はデバイスの I/O 値を含む第 5 フィールドです。

これで、前の項「[バルク・コピーの例](#)」(94 ページ) の手順に従って、*d1\_1\_device.new* ファイルを Adaptive Server テーブルにインポートできます。

*cut* ユーティリティの詳細については、システムにある UNIX のリファレンス・ページを参照してください。

## Historical Server でモニタリング・データのデータベースへの出力を可能にする

Historical Server の出力をファイルに保存するだけでなく、Historical Server からデータベースに出力を直接送信することもできます。データベースにデータを保存すると、データを分析しやすくなります。

データベースにデータを格納した場合、各 Historical Server ビューからの出力は個別のテーブルに格納されます。Historical Server ビューの定義を変更すると、テーブル定義に自動的に反映されます。

## 受信側 Adaptive Server のセットアップ

Historical Server から Adaptive Server にモニタリング・データを送信できるようにするには、送信先の Adaptive Server を選択します。Historical Server のモニタリング・データを受信する Adaptive Server を選択したら、以下を行います。

- 1 Historical Server でモニタリング・データの格納に使用するデータベースを作成します。このデータベースの名前はデフォルトで `hs_monitoring` となります。データベースに別の名前を使いたいときは、データベースを作成した後、`hs_directload.sql` スクリプト内のデータベース名を変更します。
- 2 データベースに対して `hs_directload.sql` インストール・スクリプトを実行します。このインストール・スクリプトによって、`sessions` および `views` という 2 つのカタログ・テーブルと、ストアド・プロシージャ `sp_hs_dboutput` が作成されます。

## Historical Server の起動

Historical Server から Adaptive Server のデータベースにモニタリング・データを送信できるようにするには、以下を指定します。

- Historical Server のモニタリング・データの送信先。フラット・ファイルではなく、特定の Adaptive Server とデータベースを指定します。
- 出力サーバに接続するためのユーザ名とパスワード（モニタ対象のサーバに接続するときに Historical Server が使用するものと異なる場合）。ユーザ名とパスワードを指定しない場合は、コマンド・ラインで `-U` パラメータに指定したユーザ名と `-P` パラメータに指定したパスワードがデフォルトで使用されます。

Historical Server の起動時に、履歴データのターゲット Adaptive Server とデータベースが使用可能であることが必要です。

コマンド・ラインの構文は次のとおりです。

```
histserver -U<user name>
-P<password> -D<output dir>
-l<log file> -l<interfaces file>
[-d<delimiter>] [-O<ASE name>] [-o<DATABASE name>] [-f]
-u<outputASE user name> -p<outputASEpassword>
```

- `-O ASE name` – ターゲット Adaptive Server の名前。
- `-o DATABASE name` – モニタリング・データの送信先となるデータベースの名前。データベース名が `hs_monitoring` でない場合は、このオプションでデータベース名を指定します。
- `-u outputASE user name` – 出力に使用する Adaptive Server に接続するためのログイン名。
- `-p outputASEpassword` – 出力 Adaptive Server のログイン名に対応するパスワード。

- **-f** – Historical Server からデータベースだけでなく出力ディレクトリのファイルにもデータを送信する場合、-O オプションを指定したときに使用する必要がある。

Historical Server へのアクセス権と、ターゲット・データベースに対する更新パーミッションが必要です。

次の例では、-O パラメータと -o パラメータが指定され、出力データベースは **hs\_monitoring** という名前です。

```
$SYBASE/ASE-12_5/bin/histserver -D$SYBASE -Sajax_hs
-l$SYBASE/ajax_hs.log -Usa -Pnorthstar -Oajax
-ohs_monitoring
```

## データの表示

### データの格納

Historical Server のデータは、サーバ上でビューを作成するときに定義したものと同じ形式です。データを格納するために、Historical Server は次の 2 つのシステム・テーブルを作成します。

- **sessions** テーブル。出力データベースを使用していたすべてのレコーディング・セッションが記録されます。
- **views** テーブル。各レコーディング・セッションで使用されたビューがリストされます。
- データベースに出力するビューごとに 1 つのテーブルが作成されるだけでなく、Historical Server 出力データベースに 2 つのシステム・テーブルが作成されます。これらのテーブルは次のとおりです。
  - **sessions** テーブル。出力データベースを使用していたすべてのレコーディング・セッションが記録されます。
  - **views** テーブル。各レコーディング・セッションで使用されたビューがリストされます。

次の例は、**sessions** テーブルの内容を示します。

```
1> select * from sessions
2> go
```

DbSessionId	HsSessionId	HsName	HsAseName	StartDate	EndDate
0	0	ajax_hs	ajax	Jul 22 2005 11:26AM	Jul 22 2005 11:45AM
1	14	ajax_hs	ajax	Jul 23 2005 12:25PM	Jul 23 2005 6:10AM
2	15	ajax_hs	ajax	Jul 24 2005 11:41AM	Jul 24 2005 11:59AM
3	16	ajax_hs	ajax	Jul 25 2005 11:56AM	Jul 25 2005 12:09PM
4	19	ajax_hs	ajax	Jul 26 2005 10:59AM	Jul 27 2005 11:25AM
5	23	ajax_hs	ajax	Jul 27 2005 11:26AM	Jul 27 2005 11:36AM

次の例は、views テーブルの内容を示します。

```
1> select * from views
2> go

DbSessionId ViewName
-----
1           stored_procs
1           connections
2           stored_procs_shutdown_test
2           connections
3           stored_procs_shutdown_test
3           connections
4           connections
5           stored_procedure_activity
5           connections
5           process_activity
```

ビュー・データ出力テーブルの構造は、次のように出力データ・ファイルの構造と似ています。

- ・ 第1カラムは、モニタリング・セッション ID です。
- ・ 第2カラムは、モニタ対象のサーバ名です。
- ・ 第3カラムは、日付／タイムスタンプです。
- ・ 以降のカラムは、ビュ定義で指定されたデータ項目に対応します。

日付／タイムスタンプとデータ項目については、この構造は Historical Server がデータ・ファイルを別の Adaptive Server にバルク・コピーするときに提供する DDL スクリプトに定義されている構造と同じです。

---

**注意** `viewname` はビューの名前です。現在のプラットフォームのファイル命名規則に従って、必要に応じてトランケートされます。Windows プラットフォームでは、ファイル名は 30 文字に制限されます。

---

次の例のクエリでは、2005 年 7 月 27 日にアクティブであったセッションと、それらのセッションで使用されたビューが報告されます。

```
1> select HsAseName, HsName, StartDate, EndDate, ViewName
2> from sessions, views
3> where sessions.DbSessionId = views.DbSessionId
4> and StartDate >= 'July 27, 2005'
5> and EndDate < 'July 28, 2005'
6> go

HsAseName  HsName      StartDate          EndDate          ViewName
-----
```

```

tribble  tribble_hs Jul 27 2005 10:59AM Jul 27 2005 11:00AM connections
tribble  tribble_hs Jul 27 2005 11:00AM Jul 27 2005 11:05AM sproc_activity
tribble  tribble_hs Jul 27 2005 11:00AM Jul 27 2005 11:05AM connections
tribble  tribble_hs Jul 27 2005 11:17AM Jul 27 2005 11:24AM sproc_activity
tribble  tribble_hs Jul 27 2005 11:17AM Jul 27 2005 11:24AM connections
tribble  tribble_hs Jul 27 2005 11:25AM Jul 27 2005 11:25AM sproc_activity
tribble  tribble_hs Jul 27 2005 11:25AM Jul 27 2005 11:25AM connections
tribble  tribble_hs Jul 27 2005 11:26AM Jul 27 2005 11:36AM sproc_activity
tribble  tribble_hs Jul 27 2005 11:26AM Jul 27 2005 11:36AM connections
tribble  tribble_hs Jul 27 2005 11:26AM Jul 27 2005 11:36AM process_activity

```

次に Historical Server ビューの一例を示します。

```

hs_create_view process_activity,
"Login Name", "Value for Sample",
"Process ID", "Value for Sample",
"Kernel Process ID", "Value for Sample",
"Connect Time", "Value for Session",
"Page I/O", "Value for Session",
"CPU Time", "Value for Session",
"Current Process State", "Value for Sample"

```

以下のクエリは、前述の **process\_activity** ビューによって生成されたデータベース・テーブルの内容を示します。

```

1> select * from process_activity
3> go

```

DbId	CPUTime	ProcessID	KernelProcessID	ConnectTime	PageIO	ProcessState	LogName
5	0.128447	17	3211313	61	28531	6	jsmith
5	0.128447	17	3211313	121	28531	6	jsmith
5	0.032169	19	3276850	60	32072	6	byoung
5	0.128447	17	3211313	182	28531	6	jsmith
5	0.099188	18	3342387	61	28759	6	dcharles
5	0.032169	19	3276850	121	33283	6	byoung
5	0.128447	17	3211313	242	28531	6	jsmith
5	0.105076	18	3342387	121	24373	6	dcharles
5	0.032169	19	3276850	181	33283	6	byoung
5	0.128447	17	3211313	303	28531	6	jsmith

## Historical Server セッションの削除

`hs_delete_data` コマンドを使用してレコーディング・セッションを削除した場合、デフォルトの動作では、そのセッションに対応する出力ファイルだけが削除され、出力テーブル内のエントリは元の状態のまま残されます。

出力テーブルからセッション・データを削除する場合は、次のように入力します。

```
hs_deletedata database_also, sessionId
```

このコマンドは、`Session` テーブルと `ViewPerSession` テーブル内の対応するエントリをすべて削除します。

## エラーのレポート

Historical Server は、フラット・ファイルに出力する場合と同じ方法で、モニタリング・データ出力保存プロセスでエラーを報告します。発生したエラーは、セッション・エラー・ファイルと HS ログ・ファイルによって通知されます。データベース出力モードに関連するエラーは次のとおりです。

- ターゲット Adaptive Server が存在しない場合、Historical Server は失敗を報告します。
- コマンド・ラインで指定したターゲット・データベースが存在しない場合、Historical Server は失敗を報告します。
- ディスク領域の不足によって、コマンド・ラインで指定したデータベースに書き込むことができない場合、Historical Server は失敗を報告します。
- モニタリング出力に使用するために、既存のテーブルにカラムを追加する必要がある場合、Historical Server は警告メッセージをログに記録します。

# データ項目

## データ項目および定義テーブル

表 A-1 は、レコーディング・セッション・ビューに含めることができるデータ項目の一覧です。このテーブルには、データ項目がアルファベット順に並んでおり、項目ごとに次の情報が記載されています。

- 定義
- An Adaptive Server の対応リリース
- 結果またはキー

表 A-1: データ項目および定義

データ項目	説明
Application Execution Class バージョン: 11.5 以降	<p>指定のアプリケーション名に設定された実行クラス(存在する場合)。上書き機能のために、設定された実行クラスは必ずしも実行時の優先順位とエンジン・グループを反映しません。次の表記がこのモニタで使用されます。</p> <ul style="list-style-type: none"> <li>プランク – このアプリケーションには実行クラスが設定されていないことを意味します。</li> <li>実行クラス名 – 一般的なアプリケーション用に設定された実行クラス。特定のログインを考慮しません(つまり、実行クラスは null スコープのアプリケーション用に設定されます)。</li> <li>アスタリスク (*) のついた実行クラス名 – 一般的なアプリケーション用に設定された実行クラスに加え、特定のログイン用に追加の実行クラスが設定されます(つまり、実行クラスは、null スコープのアプリケーションのために設定され、少なくとも 1 つの追加実行クラスが、特定のスコープのアプリケーションのために設定されます)。</li> <li>実行クラス名を伴わないアスタリスク (*) のついた実行クラス – 一般的なアプリケーション用に実行クラスは設定されないが、このアプリケーションを使用して、特定のログイン用の実行クラスが設定されます(つまり、少なくとも 1 つの実行クラスが、特定のスコープでアプリケーション用に設定されます)。</li> </ul> <p>タイプ: 結果</p>
Application Name バージョン: 11.0 以降	<p>他の統計値が記録されるアプリケーション名。サンプリング時間の終了時にアクティブなプロセスのみについて報告を行うアプリケーション名を含むビュー。</p> <p>ビューでは、アプリケーション名は Process ID と相互に排他的になっています。</p> <p>タイプ: キー</p>
Blocking Process ID バージョン: 11.0 以降	<p>ロックを保持し、別のプロセスを待機させているプロセスの ID(存在する場合)。ゼロの戻り値は、他のプロセスがロックされていないことを表します。</p> <p>タイプ: 結果</p>

## データ項目および定義テーブル

データ項目	説明
Cache Efficiency バージョン：11.0 以降	特定のデータ・キャッシュにおける毎秒 1MBあたりのキャッシュ・ヒット数。 タイプ：結果
Cache Hit Pct バージョン：11.0 以降	特定のデータ・キャッシュにバインドされたオブジェクトに対し、以下の計算式で求められるキャッシュが対応できるページ読み込みの割合。 $cache\_hits / (cache\_hits + cache\_misses) * 100$ タイプ：結果
Cache Hits バージョン：11.0 以降	特定のデータ・キャッシュが対応できたページ読み込みの回数。 タイプ：結果
Cache ID バージョン：11.0 以降	Adaptive Server バージョン 11.0 以降におけるデータ・キャッシュの ID。データベース・テーブルとインデックスが、それぞれ特定のデータ・キャッシュにバインドされたり、データベースのすべてのオブジェクトが同一のデータ・キャッシュにバインドされることがあります。オブジェクトは、複数のデータ・キャッシュにバインドされることはありません。 タイプ：キー
Cache Misses バージョン：11.0 以降	特定のデータ・キャッシュではなく、ディスクが対応したページ読み込みの回数。このデータ項目には、ページの割り付け中にデータ・キャッシュでのページの検索に失敗した回数が含まれます。そのため、報告された物理的なページ読み込みの回数は、多めに表示されることがあります。その場合、キャッシュ・ヒット率で報告されるデータ・キャッシュのミスの比率は、少な目に表示されます。 タイプ：結果
Cache Name バージョン：11.0 以降	データ・キャッシュの名前。データベース・テーブルとインデックスが、それぞれ特定のデータ・キャッシュにバインドされたり、データベースのすべてのオブジェクトが同一のデータ・キャッシュにバインドされることがあります。オブジェクトは、複数のキャッシュにバインドされることはありません。 タイプ：キー

データ項目	説明
Cache Prefetch Efficiency バージョン：11.0 以降	<p>大容量 I/O バッファで再使用されるページ数(分母)に対する Adaptive Server によってこれまでに参照されたページ数の割合。バッファが再使用されると、バッファ内のページはいずれも再使用としてカウントされます。データベース・デバイスからの新しい物理読み込みを受け付けるための空きバッファがプールに存在しない場合、バッファは再使用されます。Adaptive Server によって参照された再使用ページ数を、バッファごとのページ数に再使用バッファ数を乗じて求められた値で割ると、大容量 I/O フェッチの効率がわかります。</p> <p>名前付きデータ・キャッシュにバッファ・プールがどのくらいの数設定されるかに関係なく、Adaptive Server はそのうちの 2 つだけを使用します。つまり、Adaptive Server は 2K バッファ・プールと最も大きなサイズのバッファで設定されるプールを使用します。2K I/O は大容量 I/O と見なされないため、プリフェッチの効率は、2K バッファ・プールには適用されません。このため、プリフェッチ効率は、キャッシュ内の最大のバッファ・プールに適用されます。たとえば、データ・キャッシュが 2K、8K、16K のサイズのバッファのプールである場合、8K プールは使用されません。また、この測定基準は 16K バッファ・プールにある大容量 I/O の効率を反映します。</p> <p>この割合が高い場合は、プリフェッチが効率的で、低い場合は、プリフェッチによる十分な効果は期待できません。このことは、バッファ・プールが削除されるべきであることを示します(または一部のテーブルのクラスター・インデックスが断片化を起こしており、そのインデックスは削除され再作成されるべきであることを意味します)。</p> <p>タイプ：結果</p>
Cache Refer and Reuse バージョン：11.0 以降	<p>参照と再使用の両方が行われたバッファのページ数。この回数は、バッファのプリフェッチの効率を判断する際に使用します(「Cache Prefetch Efficiency」を参照)。キャッシュのプリフェッチ効率とは異なり、データ項目には、デフォルトの 2K のバッファ・プールのアクティビティが含まれます。</p> <p>バッファの再使用の定義については、「Cache Prefetch Efficiency」を参照してください。</p> <p>タイプ：結果</p>
Cache Reuse バージョン：11.0 以降	<p>再使用されたバッファのページ数。大きな値は、(メモリ・プールのバッファの)高いターンオーバー・レートを示し、そのプールが小さすぎることを意味します。ゼロの値は、このメモリ・プールが大きすぎることを示します。キャッシュのプリフェッチ効率とは異なり、データ項目には、デフォルトの 2K のバッファ・プールのアクティビティが含まれます。</p> <p>バッファの再使用の定義については、「Cache Prefetch Efficiency」を参照してください。</p> <p>タイプ：結果</p>
Cache Reuse Dirty バージョン：11.0 以降	<p>再使用されたバッファに、書き込みを必要とした変更が行われた回数。ゼロ以外の値は、ウォッシュ・サイズが小さすぎることを示します。</p> <p>バッファの再使用の定義については、「Cache Prefetch Efficiency」を参照してください。</p> <p>タイプ：結果</p>
Cache Size バージョン：11.0 以降	<p>データ・キャッシュのサイズ(MB)。</p> <p>タイプ：結果</p>
Cache Spinlock Contention バージョン：11.0 以降	<p>強制的に待機させられた、データ・キャッシュのスピンドル要求の比率。</p> <p><math>spinlock\_waits / spinlock\_requests</math></p> <p>タイプ：結果</p>

データ項目	説明
Code Memory Size	に割り当てられているメモリ・サイズ(バイト)。
バージョン: 11.0 以降	タイプ: 結果
Connect Time	プロセスまたはセッションが開始されてからの経過時間のうち、どちらか小さい方。
バージョン: 11.0 以降	タイプ: 結果
CPU Busy Percent	Adaptive Server の CPU がビジーとなった合計サーバ CPU チックの割合。
バージョン: 11.0 以降	タイプ: 結果
CPU Percent	ビューでプロセス ID とともに使用された場合、これは単一のプロセスが動作中の状態にあった時間の、すべてのプロセスが動作中の状態にあった時間に対する比率を示します。
バージョン: 11.0 以降	ビューにおいてアプリケーション名とともに使用された場合、これは、指定のアプリケーションを実行するプロセス・セットが実行状態にあった時間の、すべてのプロセスが実行状態にあった時間に対する比率を示します。
	タイプ: 結果
CPU Time	キーなしで使用した場合、このデータ項目は、サーバの CPU の合計ビジー時間の秒数を示します。キーとともに使用した場合、このデータ項目は各プロセス、アプリケーションまたはエンジンが使用したビジー時間の秒数を表します。プロセス ID とアプリケーション名は相互に排他的です。
バージョン: 11.0 以降	タイプ: 結果
CPU Yields	Adaptive Server がオペレーティング・システムに対して CPU を解放した回数。
バージョン: 11.0 以降	タイプ: 結果
Current Application Name	特定のプロセスで現在実行中のアプリケーション名。
バージョン: 11.0 以降	タイプ: 結果
Current Engine	プロセスが直前に実行されていた Adaptive Server エンジンの数。
バージョン: 11.0 以降	タイプ: 結果
Current Execution Class	プロセスが現在実行されている実行クラスの名前。
バージョン: 11.5 以降	タイプ: 結果
Current Process State	プロセスの現在のステータス。発生しうるステータスの定義については、「Process State」を参照してください。
バージョン: 11.0 以降	タイプ: 結果
Current Stmt Batch ID	特定のプロセスで実行されている、特定のクエリ・バッチの ID。
バージョン: 11.5 以降	タイプ: 結果
Current Stmt Batch Text	特定のプロセスのために実行されている、特定のクエリ・バッチのテキスト。このテキストは、クエリ・バッチのフル・テキストの先頭の部分文字列のみとなる場合があります。このフィールドに格納されるテキストの量は、Adaptive Server の <code>max SQL text monitored</code> 設定パラメータによって決まります。
バージョン: 11.5 以降	タイプ: 結果

データ項目	説明
Current Stmt Batch Text Byte Offset	特定のプロセスのために実行されているクエリ・バッチまたはストアド・プロシージャ内にある文の先頭に付けるバイト・オフセット。 バージョン：11.5 以降
	たとえば、現在の stmt プロシージャ・データベース ID と現在の stmt プロシージャ ID の両方が 0 の場合、文はクエリ・バッチで現在実行されている SQL 文です。それ以外の場合は、文はこの 2 つの ID によってユニークに識別されるストアド・プロシージャで現在実行中の SQL 文です。 タイプ：結果
Current Stmt Batch Text Enabled	Adaptive Server (バージョン 11.5 以降) が現在実行中のクエリ・バッチの SQL テキストを保存中であるかどうかを報告します。保存中の場合には、そのサイズが報告されます。 バージョン：11.5 以降
	値が 0 = SQL テキストの保存は無効。 値が 1 以上 = 保存可能なサーバ・プロセスあたりのバッチ・テキストの最大バイト数。 タイプ：結果
Current Stmt Context ID	特定のプロセスのために実行されている特定のクエリ・バッチ内のストアド・プロシージャ呼び出しをユニークに識別する ID。 バージョン：11.5 以降
	タイプ：結果
Current Stmt CPU Time	現在実行中の SQL 文が実行状態にあった合計時間の秒数。 バージョン：11.5 以降
	タイプ：結果
Current Stmt Elapsed Time	現在実行中の SQL 文が実行中であった合計時間の秒数。 バージョン：11.5 以降
	タイプ：結果
Current Stmt Line Number	特定のプロセスに現在実行中の SQL 文の先頭部分を含む (クエリ・バッチまたはストアド・プロシージャ内の) 行番号。現在の stmt プロシージャ・データベース ID と現在の stmt プロシージャ ID の両方が 0 の場合、現在実行中の SQL 文はクエリ・バッチに存在します。 バージョン：11.5 以降
	それ以外の場合は、現在実行中の SQL 文は、この 2 つの ID によってユニークに識別されるストアド・プロシージャに存在します。 タイプ：結果
Current Stmt Locks Granted After Wait	現在実行中の SQL 文によるロック要求のうち、待機後に許可されたロック要求の数。 バージョン：11.5 以降
	タイプ：結果
Current Stmt Locks Granted Immediately	現在実行中の SQL 文によるロック要求のうち、すぐに許可されたロック要求、または (リクエストによってすでに十分なロックが保持されているために) 不要となったロック要求の数。 バージョン：11.5 以降
	タイプ：結果
Current Stmt Locks Not Granted	現在実行中の SQL 文によるロック要求のうち、拒否されたロック要求の数。 バージョン：11.5 以降
	タイプ：結果
Current Stmt Logical Reads	キヤッショ読み込みまたはデバイス読み込みで対応できた、現在実行中の SQL 文によるデータ・ページ読み込みの回数。 バージョン：11.5 以降
	タイプ：結果

データ項目	説明
Current Stmt Number バージョン：11.5 以降	特定のプロセスのために現在実行中の SQL 文となる(クエリ・バッチまたはストアド・プロシージャに表示される)文の数。現在の stmt プロシージャ・データベース ID と現在の stmt プロシージャ ID の両方が 0 の場合、現在実行中の SQL 文はクエリ・バッチに存在します。 それ以外の場合は、現在実行中の SQL 文は、この 2 つの ID によってユニークに識別されるストアド・プロシージャに存在します。 値 0 は、現在実行中の SQL 文の結果データの一部を示します。つまり、この SQL 文は、モニタリングの開始前に実行を開始します。パフォーマンスの測定基準は利用可能ですが、回数はモニタリング開始後の期間のみを反映したものとなります。
	タイプ：結果
Current Stmt Page I/O バージョン：11.5 以降	現在実行中の SQL 文による、結合された論理ページ読み込みとページ書き込みの回数。
	タイプ：結果
Current Stmt Physical Reads バージョン：11.5 以降	データ・キャッシュで対応できなかった、現在実行中の SQL 文によるデータ・ページ読み込みの回数。
	タイプ：結果
Current Stmt Procedure Database ID バージョン：11.5 以降	特定のプロセスのために現在実行中の SQL 文を含むストアド・プロシージャ(特殊なストアド・プロシージャであるトリガを含む)のデータベース ID。現在実行中の SQL 文がストアド・プロシージャに含まれていない場合、この ID は 0 になります。
	タイプ：結果
Current Stmt Procedure Database Name バージョン：11.5 以降	特定のプロセスのために現在実行中の SQL 文を含むストアド・プロシージャ(特殊なストアド・プロシージャであるトリガを含む)のデータベース名。ストアド・プロシージャが現在実行中の SQL 文を含まない場合、この名前は “**NoDatabase**” になります。
	タイプ：結果
Current Stmt Procedure ID バージョン：11.5 以降	特定のプロセスのために現在実行中の SQL 文を含むストアド・プロシージャ(特殊なストアド・プロシージャであるトリガを含む)の ID。現在実行中の SQL 文がストアド・プロシージャに含まれていない場合、この ID は 0 になります。
	タイプ：結果
Current Stmt Procedure Name バージョン：11.5 以降	特定のプロセスのために現在実行中の SQL 文を含むストアド・プロシージャ(特殊なストアド・プロシージャであるトリガを含む)の名前。ストアド・プロシージャが現在実行中の SQL 文を含まない場合、この名前は “**NoObject**” になります。
	タイプ：結果
Current Stmt Procedure Owner Name バージョン：11.5 以降	特定のプロセスのために現在実行中の SQL 文を含むストアド・プロシージャ(特殊なストアド・プロシージャであるトリガを含む)の所有者名。ストアド・プロシージャが現在実行中の SQL 文を含まない場合、この名前は “**NoOwner**” になります。
	タイプ：結果
Current Stmt Procedure Text バージョン：11.5 以降	特定のプロセス用に実行中の、特定のストアド・プロシージャ(特殊なストアド・プロシージャであるトリガを含む)のテキスト。現在の stmt プロシージャ・データベース ID と現在の stmt プロシージャ ID の両方が 0 の場合、ストアド・プロシージャは現在実行されておらず、このテキストは null で終了された空の文字列 ("") となります。 (このストアド・プロシージャがコンパイルされ、そのテキストが廃棄されたか、テキストが暗号化フォーマットで格納されていることにより) テキストが利用できない場合、このテキストは null で終了された空の文字列 ("") となります。
	タイプ：結果

データ項目	説明
Current Stmt Query Plan Text	特定のプロセス用に実行されている、特定のクエリ用のクエリ・プランのテキスト。 (Adaptive Server がクエリ・プランのカタログからこのプランを削除したため) テキストが利用できない場合、このテキストは null で終了された空の文字列 ("") となります。
バージョン：11.5 以降	タイプ：結果
Current Stmt Start Time	現在実行中の SQL 文が実行を開始した Adaptive Server のタイム・ゾーンにおける日付と時間。
バージョン：11.5 以降	モニタリング開始前にこの SQL 文が実行を開始した場合、この結果は、アクティビティが最初にこの文を検出した日付と時間になります。
	タイプ：結果
Current Stmt Text Byte Offset	特定のプロセスのために実行されているクエリ・バッチまたはストアド・プロシージャ内にある文の先頭に付けるバイト・オフセット。
バージョン：11.5 以降	現在の stmt プロシージャ・データベース ID と現在の stmt プロシージャ ID の両方が 0 の場合、文はクエリ・バッチの現在実行中の SQL 文です。それ以外の場合は、文はこの 2 つの ID によってユニークに識別されるストアド・プロシージャで現在実行中の SQL 文です。
	タイプ：結果
Database ID	データベースのユニークな ID。
バージョン：11.0 以降	タイプ：キー
Database Name	データベース名
バージョン：11.0 以降	タイプ：結果
Deadlock Count	デッドロックの回数。
リリース：11.0 以降	タイプ：結果
Demand Lock	ロックがデマンド・ロック・ステータスにアップグレードされたかどうかを示す文字列 (Y または N)。
バージョン：11.0 以降	タイプ：結果
Device Hit Percent	デバイス要求の比率は、デバイス・ヒットをデバイスのミスで除算した商にデバイスのミスを加算し、それに 100 を乗算して計算します。
バージョン：11.0 以降	タイプ：結果
Device Hits	デバイスへのアクセスが許可された回数。
バージョン：11.0 以降	タイプ：結果
Device I/O	デバイス読み込みとデバイス書き込みの組み合わせ。
バージョン：11.0 以降	タイプ：結果
Device Misses	デバイスへのアクセスが待機させられた回数。
バージョン：11.0 以降	タイプ：結果
Device Name	Adaptive Server で定義されたデータベース・デバイス名。
バージョン：11.0 以降	タイプ：キー

## データ項目および定義テーブル

データ項目	説明
Device Reads	デバイスからの読み込み回数。 バージョン：11.0 以降 タイプ：結果
Device Writes	デバイスへの書き込み回数。 バージョン：11.0 以降 タイプ：結果
Elapsed Time	あるデータのリフレッシュから次の(サンプル)へ、またはビューの作成から現在の(セッション)へのいずれかの時間増分の秒数。 バージョン：11.0 以降 タイプ：結果
Engine Number	Adaptive Server エンジンの番号。 バージョン：11.0 以降 タイプ：キー
Host Name	特定の接続を確立したホスト・コンピュータの名前。 バージョン：11.0 以降 タイプ：結果
Index Logical Reads	キャッシュおよびデータベース・デバイスで対応できたインデックス・ページ読み込みの回数。 リリース：11.0 以降 タイプ：結果
Index Physical Reads	データ・キャッシュで対応できなかったインデックス・ページ読み込みの数。 バージョン：11.0 以降 タイプ：結果
Kernel Process ID	長時間ユニークのままである Adaptive Server プロセス識別子。 バージョン：11.0 以降 タイプ：キー
Kernel Structures Memory Size	カーネル構造に割り当てられているメモリ容量(バイト数)。 バージョン：11.0 以降 タイプ：結果
Large I/O Denied	複数の隣接ページをディスクから 1 度にフェッチして、このデータ・キャッシュ内のバッファにデータをロードするという(オプティマイザの)要求をバッファ・マネージャが満たさなかった回数。 バージョン：11.0 以降 タイプ：結果
Large I/O Performed	複数の隣接ページをディスクから 1 度にフェッチして、このデータ・キャッシュ内のバッファにデータをロードするという(オプティマイザの)要求をバッファ・マネージャが満たした回数。 バージョン：11.0 以降 タイプ：結果
Large I/O Requested	複数の隣接ページをディスクから 1 度にフェッチして、このデータ・キャッシュ内のバッファにデータをロードするという(バッファ・マネージャの)要求をオプティマイザが行った回数。 バージョン：11.0 以降 タイプ：結果
Lock Count	ロックの数。 バージョン：11.0 以降 タイプ：結果

データ項目	説明
Lock Hit Percent バージョン：11.0 以降	成功したロック要求の割合。 タイプ：結果
Lock Result バージョン：11.0 以降	論理ロック要求の結果。ロック結果値は、次のようにになります。 <ul style="list-style-type: none"> <li>1 – すぐに許可。</li> <li>2 – 不要 – 要求者がすでに十分なロックを保持している。</li> <li>3 – 待機 – 要求者が待機した。</li> <li>4 – 待機しなかった – ロックがすぐに取得できなかったときに、ロック要求がキューに入れられるのを要求側が待とうとしなかった。</li> <li>5 – デッドロック – 要求者がデッドロックの対象として選択された。</li> <li>6 – 割り込まれた – ロック要求がアテンション条件によって割り込まれた。</li> </ul> タイプ：キー
Lock Results Summarized バージョン：11.0 以降	許可または拒否されたレベルにおけるロック結果の要約。 <ul style="list-style-type: none"> <li>1 – 許可されたロック結果の要約は、許可、不要、待機のロック結果で構成されます。</li> <li>2 – 拒否されたロック結果の要約は、待機せず、デッドロック、中断のロック結果で構成されます。</li> </ul> タイプ：キー
Lock Status バージョン：11.0 以降	次のロック・ステータス値を含む、現在のロック・ステータス。 <ul style="list-style-type: none"> <li>1 – 保持／ブロック</li> <li>2 – 保持／ブロックなし</li> <li>3 – 要求／ブロック中</li> <li>4 – 要求／ブロックなし</li> </ul> タイプ：キー
Lock Status Count バージョン：11.0 以降	各ロック・ステータスのロック数。これはスナップショット値です。 タイプ：結果
Lock Type バージョン：11.0 以降	Adaptive Server は、アクティブなトランザクションが現在使用しているテーブルまたはデータ・ページをロックすることによって保護します。Adaptive Server は次のロック・タイプを採用しています。 <ul style="list-style-type: none"> <li>1 – 排他テーブル・ロック</li> <li>2 – 共有テーブル・ロック</li> <li>3 – 意図的排他ロック</li> <li>4 – 意図的共有ロック</li> <li>5 – 排他ページ・ロック</li> <li>6 – 共有ページ・ロック</li> <li>7 – 更新ページ・ロック</li> <li>8 – 排他ロー・ロック</li> <li>9 – 共有ロー・ロック</li> <li>10 – 更新ロー・ロック</li> </ul> タイプ：キー

データ項目	説明
Locks Being Blocked Count バージョン：11.0 以降	この “held_and_blocking” ロックを保持するプロセスによってブロックされているロック数。 タイプ：結果
Locks Granted Immediately バージョン：11.5 以降	他のロックが解除されるのを待たずに、すぐに付与されたロック数。 タイプ：結果
Locks Granted After Wait バージョン：11.5 以降	他のロックが解除されるまで待機した後に付与されたロック数。 タイプ：結果
Locks Not Granted バージョン：11.5 以降	要求されたが拒否されたロック数。 タイプ：結果
Log Contention Percent バージョン：11.0 以降	ユーザ・ログ・キャッシュがトランザクション・ログにフラッシュされた合計回数に占める、ログ・セマフォを待たなければならなかった回数の割合。 この割合が高い場合は、ユーザ・ログ・キャッシュ・サイズを大きくしてください。 タイプ：結果
Logical Page Reads バージョン：11.0 以降	キャッシュまたはデータベース・デバイスのどちらかで対応できた、単位時間あたりのデータ・ページ読み込み数。 タイプ：結果
Login Name バージョン：11.0 以降	Adaptive Server プロセスに関連するログイン名。 タイプ：結果
Most Active Device I/O バージョン：11.0 以降	所定の時間中に最もアクティブなデバイスに対する読み書き総数。 タイプ：結果
Most Active Device Name バージョン：11.0 以降	所定の時間中に読み書き総数が最大のデバイス名。 タイプ：結果
Net Bytes Received バージョン：11.0 以降	受信されたネットワーク・バイト数。 タイプ：結果
Net Bytes Sent バージョン：11.0 以降	送信されたネットワーク・バイト数。 タイプ：結果
Net Default Packet Size バージョン：11.0 以降	デフォルトのネットワーク・パケット・サイズ。 タイプ：結果
Net I/O Bytes バージョン：11.0 以降	送受信されたネットワーク・バイト総数。 タイプ：結果
Net Max Packet Size バージョン：11.0 以降	ネットワーク・パケット用に設定された最大サイズ。 タイプ：結果

データ項目	説明
Net Packet Size Received バージョン：11.0 以降	受信されたネットワーク・パケットの平均サイズ タイプ：結果
Net Packet Size Sent バージョン：11.0 以降	送信されたネットワーク・パケットの平均サイズ。 タイプ：結果
Net Packets Received バージョン：11.0 以降	受信されたネットワーク・パケット数。 タイプ：結果
Net Packets Sent バージョン：11.0 以降	送信されたネットワーク・パケット数。 タイプ：結果
Number of Engines バージョン：11.0 以降	Adaptive Server 用に設定されたエンジンの数。 タイプ：結果
Number of Processes バージョン：11.0 以降	Adaptive Server で現在実行中のプロセスの数。キー・アプリケーション名を併用している場合は、特定のアプリケーションを現在実行中のプロセスの数。 タイプ：結果
Object ID バージョン：11.0 以降	データベース・オブジェクトの ID。返されるオブジェクトは、データベース・テーブル、ストアド・プロシージャ、テンポラリ・テーブルのいずれかとなります。 オブジェクト ID は負数の場合があります。Adaptive Server がテンポラリ・テーブルに割り当てるオブジェクト ID は、正の数または負の数になります。 タイプ：キー
Object Name バージョン：11.0 以降	データベース・オブジェクト名。文字列 <b>**TempObject**</b> は、テンポラリ・テーブルで報告されます。 タイプ：結果
Object Type バージョン：11.0 以降	データベース・オブジェクトのタイプは、以下のとおりです。 <ul style="list-style-type: none"><li>0 – なし</li><li>1 – ストアド・プロシージャ (トリガを含む)</li><li>2 – テーブル</li></ul> タイプ：結果
Owner Name バージョン：11.0 以降	オブジェクトの所有者名。 タイプ：結果
Page Cache Size バージョン：11.0 以降	ページ・キャッシュに割り当てられているメモリ容量 (バイト数)。 タイプ：結果
Lock Hit Percent バージョン：11.0 以降	物理ページ読み出しを必要とせずに、キャッシュによって対応できたデータ・ページ読み込み回数の割合。 タイプ：結果
Page I/O バージョン：11.0 以降	論理ページ読み書き総数。 タイプ：結果

## データ項目および定義テーブル

データ項目	説明
Page Number	特定のロックまたはロック要求用のデータ・ページ番号。
バージョン：11.0 以降	タイプ：キー
Page Writes	データベース・デバイスに書き込まれたページの数。
バージョン：11.0 以降	タイプ：結果
Physical Page Reads	データ・キャッシュで対応できなかったデータ・ページ読み込みの数。
バージョン：11.0 以降	タイプ：結果
Procedure Buffer Size	プロシージャ・バッファに割り当てられているメモリ容量(バイト数)。
バージョン：11.0 以降	タイプ：結果
Procedure CPU Time	ストアド・プロシージャの実行に費やされた CPU 時間の秒数。
バージョン：11.0 以降	タイプ：結果
Procedure Database ID	アクティブなストアド・プロシージャのデータベース ID。
バージョン：11.0 以降	タイプ：キー
Procedure Database Name	アクティブなストアド・プロシージャのデータベース名。
バージョン：11.0 以降	タイプ：キー
Procedure Elapsed Time	ストアド・プロシージャの実行中に経過した秒数。当該データ項目報告時間(秒単位)で有効なすべての統計タイプ。たとえば、“Procedure Elapsed Time”、“Average for Session”は、各プロシージャ実行ごとに平均経過秒数を報告します。
バージョン：11.0 以降	タイプ：結果
Procedure Execution Class	ストアド・プロシージャに設定された実行クラス(存在する場合)。
バージョン：11.5 以降	タイプ：結果
Procedure Execution Class	ストアド・プロシージャまたはストアド・プロシージャの行が実行された回数。
バージョン：11.0 以降	タイプ：結果
Procedure Header Size	プロシージャ・ヘッダに割り当てられているメモリ容量(バイト数)。
バージョン：11.0 以降	タイプ：結果
Procedure Hit Percent	プロシージャの実行時に、そのプロシージャのクエリ・プランがプロシージャ・キャッシュに検出され、それが使用できる状態であった回数の割合。
バージョン：11.0 以降	タイプ：結果
Procedure ID	アクティブなストアド・プロシージャ。アクティブとは、トップ・レベルのストアド・プロシージャが呼び出されたことを意味します。
バージョン：11.0 以降	タイプ：キー
Procedure Line Number	ストアド・プロシージャの行番号。
バージョン：11.0 以降	タイプ：キー

データ項目	説明
Procedure Line Text	ストアド・プロシージャのテキスト全体。 バージョン：11.0 以降 タイプ：結果
Procedure Logical Reads	ストアド・プロシージャの実行要求回数。プロシージャ・キャッシュが対応した場合と <i>sysprocedures</i> からの読み込みで対応した場合のいずれか。 バージョン：11.0 以降 タイプ：結果
Procedure Name	アクティブなストアド・プロシージャ名。 バージョン：11.0 以降 タイプ：結果
Procedure Owner Name	アクティブなストアド・プロシージャの所有者名。 バージョン：11.0 以降 タイプ：結果
Procedure Physical Reads	<i>sysprocedures</i> からの読み込みが必要となった、ストアド・プロシージャの実行要求回数。 バージョン：11.0 以降 タイプ：結果
Procedure Statement Number	ストアド・プロシージャ内の文の番号。ストアド・プロシージャの 1 行には、1 つ以上の文を設定することができます。 バージョン：11.0 以降 タイプ：キー
Process ID	Adaptive Server プロセス識別番号。プロセス ID が含まれるビューは、サンプリング時間が終了した時点でアクティブなプロセスについてのみ報告します。ビューでは、プロセス ID はアプリケーション名と相互に排他的になっています。 バージョン：11.0 以降 タイプ：キー
Process State	プロセス・ステータスは以下のとおりです。 バージョン：11.0 以降 <ul style="list-style-type: none"> <li>0 – なし</li> <li>1 – アラーム待ち状態。アラームを待機中。</li> <li>2 – バックグラウンド。Adaptive Server プロセスが実行中。</li> <li>3 – ステータス不良。不明なエラー条件。</li> <li>4 – 重大エラー。Adaptive Server が処理不能としてマーク付けした。</li> <li>5 – ロック取得待ち状態。ロック取得の待機中。</li> <li>6 – スリープ中。ネットワーク読み込みの待機中。</li> <li>7 – 実行可能。優先度と CPU の可用性に従って実行待ち。</li> <li>8 – 実行中。実行している。</li> <li>9 – 送信待ち状態。ネットワーク送信の待機中。</li> <li>10 – スリープ中。デバイス I/O (物理読み込み) 待ちやクライアントのアクティビティ待ちなど、ここにリストされていない他の理由で一時停止中。</li> <li>11 – 停止。プロセスは停止した。</li> <li>12 – 終了。プロセスを終了している。</li> <li>13 – 不明。不明なプロセス・ステータス。</li> <li>14 – リモート I/O。リモート (OMNI) サーバによるオペレーションの終了を待機中。</li> <li>15 – 同期メッセージ待ち状態。クエリを並列で実行する他のサーバ・プロセスとの同期を待っている。</li> </ul>

## データ項目および定義テーブル

データ項目	説明
Process State Count	特定のステータスにあるプロセス数。
バージョン：11.0 以降	タイプ：結果
Rows Deleted	データベース・テーブルから削除されたローの数。
バージョン：11.0 以降	タイプ：結果
Rows Deleted Deferred	データベース・テーブルから遅延モードで削除されたローの数。
バージョン：11.0 以降	タイプ：結果
Rows Deleted Direct	データベース・テーブルから直接モードで削除されたローの数。
バージョン：11.0 以降	タイプ：結果
Rows Inserted	データベース・テーブルへの挿入。
バージョン：11.0 以降	タイプ：結果
Rows Inserted Clustered	クラスタード・インデックスを使用したデータベース・テーブルへの挿入。
バージョン：11.0 以降	タイプ：結果
Rows Inserted Heap	クラスタード・インデックスを使用しないデータベース・テーブルへの挿入。
バージョン：11.0 以降	タイプ：結果
Rows Updated	データベース・テーブルの更新。
バージョン：11.0 以降	タイプ：結果
Rows Updated Deferred	完了まで 2 ステップを要する更新。最初に既存のエントリを削除し、新しいエントリを挿入するログ・レコードが書き込まれますが、実際にはデータ・ページでは削除のみが行われます。2 番目の段階では、ログが再スキャンされ、データ・ページに対して挿入オペレーションが実行されます。
バージョン：11.0 以降	タイプ：結果
Rows Updated Direct	高コストの更新、置き換え更新、非置き換え更新の合計(それぞれ遅延更新を除く)。
バージョン：11.0 以降	タイプ：結果
Rows Updated Expensive	ローのオリジナルの位置からローを削除し、次にそのローを新しい位置に挿入する直接更新のタイプ。
バージョン：11.0 以降	
Rows Updated In Place	ローがデータ・ページに移動する必要のない直接更新のタイプ。
バージョン：11.0 以降	タイプ：結果
Rows Updated Not In Place	更新されたローが移動する必要のない直接更新のタイプ。しかし、更新ローの長さが変更になるため、データ・ページの他のローは移動します。低コストの更新ともいいます。
バージョン：11.0 以降	タイプ：結果
Select Statements	SELECT 文または OPEN CURSOR 文の数。
バージョン：11.0 以降	タイプ：結果

データ項目	説明
Server Structures Size バージョン：11.0 以降	Adaptive Server 構造に割り当てられているメモリ・サイズ (バイト数)。 タイプ：結果
SQL Server Name バージョン：11.0 以降	使用中の Monitor Server の起動コマンドの -S パラメータで指定した方法でモニタリングを行っている Adaptive Server の名前。 タイプ：結果
SQL Server Version バージョン：11.0 以降	モニタしている Adaptive Server のバージョン。詳細については、『ASE Transact-SQL ユーザーズ・ガイド』の「 <code>@@version</code> グローバル変数」を参照してください。 タイプ：結果
Thread Exceeded Max バージョン：11.5 以降	Adaptive Server バージョン 11.5 以降において、サーバ全体のワーカ・スレッド・プールの上限スレッド数を超えたため、実行時にクエリ・プランが調整された回数。 タイプ：結果
Thread Exceeded Max Percent バージョン：11.5 以降	Adaptive Server バージョン 11.5 以降において、サーバ全体のワーカ・スレッド・プールの上限スレッド数を超えたため、実行時にクエリ・プランが調整された回数の比率。 タイプ：結果
Thread Max Used バージョン：11.5 以降	サーバ上で同時に使用されていた、サーバ全体のワーカ・スレッド・プールからのスレッドの最大数。 タイプ：結果
Time Waited on Lock バージョン：11.0 以降	ロック要求が待機状態になっていた合計時間の秒数。 タイプ：結果
Timestamp バージョン：11.0 以降	<code>summarization_level</code> が raw の場合のレコーディング・セッション・ビューとプレイバック・ビューで、レコーディング・セッション・データが収集されたときの Adaptive Server における日付と時刻。 プレイバック・ビューでは、 <code>summarization_level</code> が actual、entire、またはユーザ定義の間隔の場合、時間は Historical Server のタイム・ゾーンに変換されます。 詳細については、『ASE Transact-SQL ユーザーズ・ガイド』の「 <code>getdate()</code> 関数」を参照してください。 タイプ：結果
Timestamp Datim バージョン：11.0 以降	<code>summarization_level</code> が raw の場合のレコーディング・セッション・ビューとプレイバック・ビューで、レコーディング・セッション・データが収集され、CS_DATETIME 構造体に返されたときの Adaptive Server における日付と時刻。詳細については、『Transact-SQL ユーザーズ・ガイド』の「 <code>getdate()</code> 関数」を参照してください。 プレイバック・ビューでは、 <code>summarization_level</code> が actual、entire、またはユーザ定義の間隔の場合、時間は Historical Server のタイム・ゾーンに変換されます。 タイプ：結果
Transactions バージョン：11.0 以降	<code>begin transaction</code> 文と <code>commit transaction</code> 文で区切られた、コミットされた Transact-SQL 文ブロックの総数。 タイプ：結果



# レコーディング・セッション・ビューの定義の仕様

この付録では、レコーディング・セッション・ビューを設計するための規則と考慮事項について説明します。

トピック名	ページ
<a href="#">キーと結果の定義</a>	119
<a href="#">レコーディング・セッション・ビューの設計</a>	120
<a href="#">有効なキー・データ項目と結果データ項目の組み合わせ表</a>	122
<a href="#">各データ項目に対して有効な統計タイプの表</a>	145

## キーと結果の定義

データ項目は、キーと結果のいずれかに分類されます。

- キー・データ項目は、ビューに返されるローをユニークに識別します。たとえば、プロセス単位のデータを定義するビューでは、[Process ID] データ項目が各プロセスを識別します。
- 結果データ項目は、各キー値に関連して返される情報を識別します。

キーと結果を組み合わせてレコーディング・セッション・ビューを定義することで、Historical Server から返されるデータを制御します。

ビューではキー・データ項目を組み合わせて、返されるデータの範囲を絞ることができます。連続させるそれぞれのキーとともに、ビューの定義に “per” という単語を追加します。たとえば、ページ I/O を、データベース単位、プロセス単位というように絞っていきます (データ項目ごとに有効なキーの組み合わせが決まっています。表 B-2 (122 ページ) を参照してください)。

サーバ全体のデータを返すには、キーを含まないビューを定義します。キーを含まないビューでは、結果は常に 1 つのローで構成されます。(サーバ・レベルで有効なデータ項目は決まっています。表 B-2 (122 ページ) を参照してください)。

[表 B-1](#) に、さまざまなキーの組み合わせを使用したビューの例を示します。

表 B-1: さまざまなキーの組み合わせを使用したビューの例

ビュー定義	結果
Page I/O	ビューにはキーが含まれていないため、サーバ全体のページ I/O が返される。例: Page I/O ----- 145
Process ID (キー)、Login Name、Page I/O	プロセス単位のページ I/O が返される。例: Process ID      Login Name      Page I/O ----- 1                    sa                45 5                    joe              100
Process ID (キー)、Database ID (キー)、Object ID (キー)、Database Name、Object Name、Page I/O	プロセス単位かつデータベース・テーブル単位のページ I/O が返される。例: Process ID      Database ID      Object ID      Database Name      Object Name      Page I/O ----- 1                    5                208003772    pubs2    authors        10 1                    5                336004228    pubs2    titles         35 5                    5                208003772    pubs2    authors        100

## レコーディング・セッション・ビューの設計

ビューを定義するには、次の手順に従います。

- [表 B-2 \(122 ページ\)](#) を使用して、キー・データ項目と結果データ項目の有効な組み合わせを選択します。
- [表 B-3 \(145 ページ\)](#) を使用して、データ項目ごとに有効な統計タイプを選択します。

## Process ID の使用

サーバ・プロセスが終了すると、Adaptive Server はそのプロセス ID を新しいプロセスのために再使用することがあります。したがって、[Process ID] データ項目では、プロセスをユニークに識別できるとは限りません。ただし、[Kernel Process ID] データ項目を使用すると、プロセスをユニークに識別できます。

プロセス単位のデータを表示する、raw でないプレイバック・ビューを作成する場合、レコーディング・セッション・ビューとプレイバック・ビューには、[Process ID] と [Kernel Process ID] の両方のデータ項目が必要です。プレイバックが raw の場合、使用できるのは [Process ID] だけです。

Process ID を含むビューは、次のようにローを返します。

- レコーディング・セッション・ビューと raw プレイバック・ビューの場合これらのビューは、サンプル間隔の最後に存在した Process ID のローだけを返します。サンプル間隔の途中でサーバ・プロセスが終了した場合、その Process ID のローは返されません。
- セッションの *summarization\_level* パラメータが *actual*、*entire*、またはユーザ定義の間隔であるプレイバック・ビューの場合これらのビューは、入力されたサンプルのいずれかに含まれているサーバ・プロセスすべてのローを返します。ただし、プロセス ID がユニークであるという保証はないため、キーの一意性を保証するには [Kernel Process ID] データ項目も使用します。このデータ項目がないと、誤って 2 つの異なるプロセスが要約される可能性があります。

## アプリケーション名の使用

Historical Server は、同じアプリケーション名を持つすべてのプロセスのデータを合計することで、アプリケーション単位のパフォーマンス・データを蓄積します。また、サンプル間隔の最後の時点で存在したプロセスのパフォーマンス・データだけを収集します。詳細については、「[Process ID の使用](#)」(120 ページ) を参照してください。

## ビューに空のローがある場合とローがない場合

報告するアクティビティがないとき、空のロー(結果データ項目に対する値を持たないロー)を返すデータ項目と、ローを省略するデータ項目があります。ビューに空のローが表示されるかどうかを決めている規則は次のとおりです。

- サーバ・レベルのデータ項目は、報告するアクティビティがない場合も常にローを返す
- キー・データ項目 [Process ID] または [Application Name] を含むビューは、サンプル間隔の最後の時点でアクティブなプロセスだけを報告する
- キー・データ項目 [Object ID] または [Stored Procedure ID] を含むビューは、サンプル間隔で報告するアクティビティがない場合にローを省略する
- 上記のデータ項目以外のキーを含むビューは、アクティビティがなかった場合もローを返す

## 有効なキー・データ項目と結果データ項目の組み合わせ表

ビュー内で組み合わせができるデータ項目には制約があります。制約は、データ項目間の関係に基づいて決まります。たとえば、“Page Number” 単位の “CPU Busy Percent” を調べても意味がありません。

表 B-2 は、ビューの結果データ項目とキー・データ項目の有効な組み合わせを示します。表には、結果データ項目がアルファベット順にリストされています。ある結果データ項目をレコードイング・セッション・ビューに指定できるかどうかを判断するための基本的な規則は次のとおりです。

- ビューには、結果データ項目にリストされている必須キーをすべて指定する必要があります。
- ビューには、結果データ項目にリストされているオプション・キーを指定できます。ただし、データ項目にリストされているオプション・キーを複数指定した場合、それらがともに機能するとは限りません。
- ビューには、必須またはオプションとしてリストされているキー以外のキーは指定できません(例外については、表の注で説明しています)。
- 「サーバ・レベルで有効か」は、キーを含まないビューでこのデータ項目を使用して、Adaptive Server に関する一覧または静的情報を取得できるかどうかを示します。
- 必須キーを持たない結果データ項目は、サーバ・レベル・ビューで使用できます。サーバ・レベル・ビューにはキーはなく、Adaptive Server に関する一覧または静的情報が報告されます。
- 脚注は、ビューで結果データ項目を使用するための追加情報です。

表 B-2: 有効なキーと結果の組み合わせ

データ項目	必須キーとオプション・キー
Application Execution Class <sup>2</sup>	サーバ・レベルで有効か いいえ 必須キー： Application Name
Application Name (キー) <sup>3</sup>	
Blocking Process ID	サーバ・レベルで有効か いいえ 必須キー： • Process ID • Database ID • Object ID • Lock Status • Page Number オプション・キー： Lock Type

データ項目	必須キーとオプション・キー
Cache Efficiency	サーバ・レベルで有効か いいえ 必須キー： Cache ID
Cache Hit Pct	サーバ・レベルで有効か いいえ 必須キー： Cache ID
Cache Hits	サーバ・レベルで有効か いいえ 必須キー： Cache ID
Cache ID (キー) <sup>4</sup>	
Cache Misses	サーバ・レベルで有効か いいえ 必須キー： Cache ID
Cache Name <sup>4</sup>	サーバ・レベルで有効か いいえ 必須キー： Cache ID
Cache Prefetch Efficiency	サーバ・レベルで有効か いいえ 必須キー： Cache ID
Cache Refer and Reuse	サーバ・レベルで有効か いいえ 必須キー： Cache ID
Cache Reuse	サーバ・レベルで有効か いいえ 必須キー： Cache ID

## 有効なキー・データ項目と結果データ項目の組み合わせ表

データ項目	必須キーとオプション・キー
Cache Reuse Dirty	サーバ・レベルで有効か はい 必須キー： なし オプション・キー： Cache ID
Cache Size	サーバ・レベルで有効か いいえ 必須キー： Cache ID
Cache Spinlock Contention	サーバ・レベルで有効か いいえ 必須キー： Cache ID
Code Memory Size	サーバ・レベルで有効か はい 必須キー： なし
Connect Time	サーバ・レベルで有効か いいえ 必須キー： Process ID
CPU Busy Percent	サーバ・レベルで有効か はい 必須キー： なし オプション・キー： Engine Number
CPU Percent <sup>3</sup>	サーバ・レベルで有効か いいえ 必須キー： Process ID または Application Name オプション・キー： Engine Number

データ項目	必須キーとオプション・キー
CPU Time <sup>3</sup>	<p>サーバ・レベルで有効か はい 必須キー： なし オプション・キー： • Process ID または • Application Name • Engine Number</p>
CPU Yields	<p>サーバ・レベルで有効か はい 必須キー： なし オプション・キー： Engine Number</p>
Current Application Name	<p>サーバ・レベルで有効か いいえ 必須キー： Process ID</p>
Current Engine	<p>サーバ・レベルで有効か いいえ 必須キー： Process ID</p>
Current Execution Class <sup>2</sup>	<p>サーバ・レベルで有効か いいえ 必須キー： Process ID</p>
Current Process State	<p>サーバ・レベルで有効か いいえ 必須キー： Process ID</p>
Current Stmt Batch ID <sup>2</sup>	<p>サーバ・レベルで有効か いいえ 必須キー： Process ID</p>
Current Stmt Batch Text <sup>2</sup>	<p>サーバ・レベルで有効か いいえ 必須キー： Process ID</p>

データ項目	必須キーとオプション・キー
Current Stmt Batch Text Byte Offset <sup>2</sup>	サーバ・レベルで有効か いいえ 必須キー： Process ID
Current Stmt Batch Text Enabled <sup>2</sup>	サーバ・レベルで有効か はい 必須キー： なし
Current Stmt Context ID <sup>2</sup>	サーバ・レベルで有効か いいえ 必須キー： Process ID
Current Stmt CPU Time <sup>2</sup>	サーバ・レベルで有効か いいえ 必須キー： Process ID
Current Stmt Elapsed Time <sup>2</sup>	サーバ・レベルで有効か いいえ 必須キー： Process ID
Current Stmt Line Number <sup>2</sup>	サーバ・レベルで有効か いいえ 必須キー： Process ID
Current Stmt Locks Granted After Wait <sup>2</sup>	サーバ・レベルで有効か いいえ 必須キー： Process ID
Current Stmt Locks Granted Immediately <sup>2</sup>	サーバ・レベルで有効か いいえ 必須キー： Process ID
Current Stmt Locks Not Granted <sup>2</sup>	サーバ・レベルで有効か いいえ 必須キー： Process ID

データ項目	必須キーとオプション・キー
Current Stmt Logical Reads	サーバ・レベルで有効か いいえ 必須キー： Process ID
Current Stmt Number <sup>2</sup>	サーバ・レベルで有効か いいえ 必須キー： Process ID
Current Stmt Page I/O <sup>2</sup>	サーバ・レベルで有効か いいえ 必須キー： Process ID
Current Stmt Page Writes <sup>2</sup>	サーバ・レベルで有効か いいえ 必須キー： Process ID
Current Stmt Physical Reads <sup>2</sup>	サーバ・レベルで有効か いいえ 必須キー： Process ID
Current Stmt Procedure Database ID <sup>2</sup>	サーバ・レベルで有効か いいえ 必須キー： Process ID
Current Stmt Procedure Database Name <sup>2</sup>	サーバ・レベルで有効か いいえ 必須キー： Process ID
Current Stmt Procedure ID <sup>2</sup>	サーバ・レベルで有効か いいえ 必須キー： Process ID
Current Stmt Procedure Name <sup>2</sup>	サーバ・レベルで有効か いいえ 必須キー： Process ID

データ項目	必須キーとオプション・キー
Current Stmt Procedure Owner Name <sup>2</sup>	サーバ・レベルで有効か いいえ 必須キー： Process ID
Current Stmt Procedure Text <sup>2</sup>	サーバ・レベルで有効か いいえ 必須キー： Process ID
Current Stmt Query Plan Text <sup>2</sup>	サーバ・レベルで有効か いいえ 必須キー： Process ID
Current Stmt Start Time <sup>2</sup>	サーバ・レベルで有効か いいえ 必須キー： Process ID
Current Stmt Text Byte Offset <sup>2</sup>	サーバ・レベルで有効か いいえ 必須キー： Process ID
Database ID (キー)	
Database Name <sup>1</sup>	サーバ・レベルで有効か いいえ 必須キー： Database ID
Deadlock Count	サーバ・レベルで有効か はい 必須キー： なし

データ項目	必須キーとオプション・キー
Demand Lock	<p>サーバ・レベルで有効か いいえ 必須キー：<ul style="list-style-type: none"><li>• Process ID</li><li>• Database ID</li><li>• Object ID</li><li>• Lock Status</li><li>• Page Number</li></ul>オプション・キー： Lock Type</p>
Device Hit Percent	<p>サーバ・レベルで有効か はい 必須キー： なし オプション・キー： Device Name</p>
Device Hits	<p>サーバ・レベルで有効か はい 必須キー： なし オプション・キー： Device Name</p>
Device I/O	<p>サーバ・レベルで有効か はい 必須キー： なし オプション・キー： Device Name</p>
Device Misses	<p>サーバ・レベルで有効か はい 必須キー： なし オプション・キー： Device Name</p>
Device Name (キー)	

## 有効なキー・データ項目と結果データ項目の組み合わせ表

データ項目	必須キーとオプション・キー
Device Reads	サーバ・レベルで有効か はい 必須キー： なし オプション・キー： Device Name
Device Writes	サーバ・レベルで有効か はい 必須キー： なし オプション・キー： Device Name
Elapsed Time	サーバ・レベルで有効か はい 必須キー： なし
Engine Number (キー)	
Host Name	サーバ・レベルで有効か いいえ 必須キー： Process ID
Index Logical Reads	サーバ・レベルで有効か はい 必須キー： なし オプション・キー： <ul style="list-style-type: none"> <li>• Process ID</li> <li>• Database ID</li> <li>• Object ID</li> <li>• Engine Number</li> <li>• キーの組み合わせ：               <ul style="list-style-type: none"> <li>• Procedure Database ID</li> <li>• Procedure ID</li> </ul> </li> </ul>

データ項目	必須キーとオプション・キー
Index Physical Reads	<p>サーバ・レベルで有効か はい 必須キー： なし オプション・キー：  <ul style="list-style-type: none"> <li>• Process ID</li> <li>• Database ID</li> <li>• Object ID</li> <li>• Engine Number</li> <li>• キーの組み合わせ：  <ul style="list-style-type: none"> <li>• Procedure Database ID</li> <li>• Procedure ID</li> </ul> </li> </ul> </p>
Kernel Process ID <sup>1</sup>	<p>サーバ・レベルで有効か いいえ 必須キー： Process ID</p>
Kernel Structures Memory Size	<p>サーバ・レベルで有効か はい 必須キー： なし</p>
Large I/O Denied	<p>サーバ・レベルで有効か いいえ 必須キー： Cache ID</p>
Large I/O Performed	<p>サーバ・レベルで有効か いいえ 必須キー： Cache ID</p>
Large I/O Requested	<p>サーバ・レベルで有効か いいえ 必須キー： Cache ID</p>

データ項目	必須キーとオプション・キー
Lock Count	サーバ・レベルで有効か はい 必須キー： なし オプション・キー： • Process ID • Lock Type • Lock Result • Lock Results Summarized
Lock Hit Percent	サーバ・レベルで有効か はい 必須キー： なし
Lock Result (キー)	
Lock Results Summarized (キー)	
Lock Status (キー)	
Lock Status Count	サーバ・レベルで有効か いいえ 必須キー： Lock Status
Lock Type (キー)	
Locks Being Blocked Count	サーバ・レベルで有効か いいえ 必須キー： • Process ID • Database ID • Object ID • Lock Status • Lock Type • Page Number

データ項目	必須キーとオプション・キー
Locks Granted After Wait <sup>3</sup>	<p>サーバ・レベルで有効か はい 必須キー： なし オプション・キー：  <ul style="list-style-type: none"> <li>• Process ID または</li> <li>• Application Name</li> <li>• キーの組み合わせ：  <ul style="list-style-type: none"> <li>• Database ID</li> <li>• Object ID</li> </ul> </li> <li>• キーの組み合わせ：  <ul style="list-style-type: none"> <li>• Procedure Database ID</li> <li>• Procedure ID</li> </ul> </li> </ul> </p>
Locks Granted Immediately <sup>3</sup>	<p>サーバ・レベルで有効か はい 必須キー： なし オプション・キー：  <ul style="list-style-type: none"> <li>• Process ID または</li> <li>• Application Name</li> <li>• キーの組み合わせ：  <ul style="list-style-type: none"> <li>• Database ID</li> <li>• Object ID</li> </ul> </li> <li>• キーの組み合わせ：  <ul style="list-style-type: none"> <li>• Procedure Database ID</li> <li>• Procedure ID</li> </ul> </li> </ul> </p>

データ項目	必須キーとオプション・キー
Locks Not Granted <sup>3</sup>	<p>サーバ・レベルで有効か はい 必須キー： なし オプション・キー：  <ul style="list-style-type: none"> <li>• Process ID または</li> <li>• Application Name</li> <li>• キーの組み合わせ：  <ul style="list-style-type: none"> <li>• Database ID</li> <li>• Object ID</li> </ul> </li> <li>• キーの組み合わせ：  <ul style="list-style-type: none"> <li>• Procedure Database ID</li> <li>• Procedure ID</li> </ul> </li> </ul> </p>
Log Contention Percent	<p>サーバ・レベルで有効か はい 必須キー： なし</p>
Logical Page Reads	<p>サーバ・レベルで有効か はい 必須キー： なし オプション・キー：  <ul style="list-style-type: none"> <li>• Process ID</li> <li>• Engine Number</li> <li>• キーの組み合わせ：  <ul style="list-style-type: none"> <li>• Database ID</li> <li>• Object ID</li> </ul> </li> <li>• キーの組み合わせ：  <ul style="list-style-type: none"> <li>• Procedure Database ID</li> <li>• Procedure ID</li> </ul> </li> </ul> </p>
Login Name <sup>1</sup>	<p>サーバ・レベルで有効か いいえ 必須キー： Process ID</p>
Most Active Device I/O	<p>サーバ・レベルで有効か はい 必須キー： なし</p>

データ項目	必須キーとオプション・キー
Most Active Device Name	サーバ・レベルで有効か はい 必須キー： なし
Net Bytes Received	サーバ・レベルで有効か はい 必須キー： なし
Net Bytes Sent	サーバ・レベルで有効か はい 必須キー： なし
Net Default Packet Size	サーバ・レベルで有効か はい 必須キー： なし
Net I/O Bytes	サーバ・レベルで有効か はい 必須キー： なし
Net Max Packet Size	サーバ・レベルで有効か はい 必須キー： なし
Net Packet Size Received	サーバ・レベルで有効か はい 必須キー： なし
Net Packet Size Sent	サーバ・レベルで有効か はい 必須キー： なし
Net Packets Received	サーバ・レベルで有効か はい 必須キー： なし

データ項目	必須キーとオプション・キー
Net Packets Sent	サーバ・レベルで有効か はい 必須キー： なし
Number of Engines	サーバ・レベルで有効か はい 必須キー： なし
Number of Processes <sup>2</sup>	サーバ・レベルで有効か はい 必須キー： なし オプション・キー： Application Name
Object ID(キー)	
Object Name <sup>1</sup>	サーバ・レベルで有効か いいえ 必須キー： • Database ID • Object ID
Object Type <sup>1</sup>	サーバ・レベルで有効か いいえ 必須キー： • Database ID • Object ID
Owner Name <sup>1</sup>	サーバ・レベルで有効か いいえ 必須キー： • Database ID • Object ID
Page Cache Size	サーバ・レベルで有効か はい 必須キー： なし

データ項目	必須キーとオプション・キー
Lock Hit Percent	<p>サーバ・レベルで有効か はい 必須キー： なし オプション・キー：  <ul style="list-style-type: none"> <li>• Process ID</li> <li>• Engine Number</li> </ul> </p> <p>キーの組み合わせ：  <ul style="list-style-type: none"> <li>• Database ID</li> <li>• Object ID</li> </ul> </p> <p>キーの組み合わせ：  <ul style="list-style-type: none"> <li>• Procedure Database ID</li> <li>• Procedure ID</li> </ul> </p>
Page I/O	<p>サーバ・レベルで有効か はい 必須キー： なし オプション・キー：  <ul style="list-style-type: none"> <li>• Process ID</li> <li>• Engine Number</li> </ul> </p> <p>キーの組み合わせ：  <ul style="list-style-type: none"> <li>• Database ID</li> <li>• Object ID</li> </ul> </p> <p>キーの組み合わせ：  <ul style="list-style-type: none"> <li>• Procedure Database ID</li> <li>• Procedure ID</li> </ul> </p>
Page Number(キー )	

データ項目	必須キーとオプション・キー
Page Writes	<p>サーバ・レベルで有効か はい 必須キー： なし オプション・キー：  <ul style="list-style-type: none"> <li>• Process ID</li> <li>• Engine Number</li> <li>• キーの組み合わせ：  <ul style="list-style-type: none"> <li>• Database ID</li> <li>• Object ID</li> </ul> </li> <li>• キーの組み合わせ：  <ul style="list-style-type: none"> <li>• Procedure Database ID</li> <li>• Procedure ID</li> </ul> </li> </ul> </p>
Physical Page Reads	<p>サーバ・レベルで有効か はい 必須キー： なし オプション・キー：  <ul style="list-style-type: none"> <li>• Process ID</li> <li>• Engine Number</li> <li>• キーの組み合わせ：  <ul style="list-style-type: none"> <li>• Database ID</li> <li>• Object ID</li> </ul> </li> <li>• キーの組み合わせ：  <ul style="list-style-type: none"> <li>• Procedure Database ID</li> <li>• Procedure ID</li> </ul> </li> </ul> </p>
Procedure Buffer Size	<p>サーバ・レベルで有効か はい 必須キー： なし</p>

データ項目	必須キーとオプション・キー
Procedure CPU Time	<p>サーバ・レベルで有効か いいえ 必須キー：<ul style="list-style-type: none"><li>Procedure Database ID</li><li>Procedure ID</li></ul>オプション・キー：<ul style="list-style-type: none"><li>Process ID</li><li>Procedure Stmt Number</li><li>Procedure Line Number</li></ul></p>
Procedure Database ID (キー) Procedure Database Name <sup>1</sup>	<p>サーバ・レベルで有効か いいえ 必須キー： Procedure Database ID</p>
Procedure Elapsed Time	<p>サーバ・レベルで有効か いいえ 必須キー：<ul style="list-style-type: none"><li>Procedure Database ID</li><li>Procedure ID</li></ul>オプション・キー：<ul style="list-style-type: none"><li>Process ID</li><li>Procedure Stmt Number</li><li>Procedure Line Number</li></ul></p>
Procedure Execution Class <sup>2</sup>	<p>サーバ・レベルで有効か いいえ 必須キー：<ul style="list-style-type: none"><li>Procedure Database ID</li><li>Procedure ID</li></ul></p>
Procedure Execution Class	<p>サーバ・レベルで有効か いいえ 必須キー：<ul style="list-style-type: none"><li>Procedure Database ID</li><li>Procedure ID</li></ul>オプション・キー：<ul style="list-style-type: none"><li>Process ID</li><li>Procedure Stmt Number</li><li>Procedure Line Number</li></ul></p>

## 有効なキー・データ項目と結果データ項目の組み合わせ表

データ項目	必須キーとオプション・キー
Procedure Header Size	サーバ・レベルで有効か はい 必須キー： なし
Procedure Hit Percent	サーバ・レベルで有効か はい 必須キー： なし
Procedure ID (キー)	
Procedure Line Number (キー)	
Procedure Line Text	サーバ・レベルで有効か いいえ 必須キー： • Procedure Database ID • Procedure ID
Procedure Logical Reads	サーバ・レベルで有効か はい 必須キー： なし
Procedure Name <sup>1</sup>	サーバ・レベルで有効か いいえ 必須キー： • Procedure Database ID • Procedure ID
Procedure Owner Name <sup>1</sup>	サーバ・レベルで有効か いいえ 必須キー： • Procedure Database ID • Procedure ID
Procedure Physical Reads	サーバ・レベルで有効か はい 必須キー： なし
Procedure Statement Number (キー)	
Process ID (キー) <sup>3</sup>	
Process State (キー)	

データ項目	必須キーとオプション・キー
Process State Count	サーバ・レベルで有効か いいえ 必須キー： Process State
Rows Deleted	サーバ・レベルで有効か はい 必須キー： なし
Rows Deleted Deferred	サーバ・レベルで有効か はい 必須キー： なし
Rows Deleted Direct	サーバ・レベルで有効か はい 必須キー： なし
Rows Inserted	サーバ・レベルで有効か はい 必須キー： なし
Rows Inserted Clustered	サーバ・レベルで有効か はい 必須キー： なし
Rows Inserted Heap	サーバ・レベルで有効か はい 必須キー： なし
Rows Updated	サーバ・レベルで有効か はい 必須キー： なし
Rows Updated Deferred	サーバ・レベルで有効か はい 必須キー： なし

## 有効なキー・データ項目と結果データ項目の組み合わせ表

データ項目	必須キーとオプション・キー
Rows Updated Direct	サーバ・レベルで有効か はい 必須キー： なし
Rows Updated Expensive	サーバ・レベルで有効か はい 必須キー： なし
Rows Updated In Place	サーバ・レベルで有効か はい 必須キー： なし
Rows Updated Not In Place	サーバ・レベルで有効か はい 必須キー： なし
Select Statements	サーバ・レベルで有効か はい 必須キー： なし
Server Structures Size	サーバ・レベルで有効か はい 必須キー： なし
SQL Server Name	サーバ・レベルで有効か はい 必須キー： なし
SQL Server Version	サーバ・レベルで有効か はい 必須キー： なし
Thread Exceeded Max <sup>2</sup>	サーバ・レベルで有効か はい 必須キー： なし

データ項目	必須キーとオプション・キー
Thread Exceeded Max Percent <sup>2</sup>	サーバ・レベルで有効か はい 必須キー： なし
Thread Max Used <sup>2</sup>	サーバ・レベルで有効か はい 必須キー： なし
Time Waited on Lock	サーバ・レベルで有効か いいえ 必須キー： • Process ID • Database ID • Object ID • Lock Status • Page Number オプション・キー： Lock Type
Timestamp	サーバ・レベルで有効か はい 必須キー： なし
Timestamp Datim	サーバ・レベルで有効か はい 必須キー： なし
Transactions	サーバ・レベルで有効か はい 必須キー： なし

データ項目	必須キーとオプション・キー
1.	これらのデータ項目は、ID またはステータス・タイプから名前を導出します。これらは、必須キーを含むすべてのビューで有効です。このことは、そのビューで他にどのキーが使用されているかに関係しません（たとえば、[Database Name] は、キー [Database ID] を含むすべてのビューで有効です。したがって、キーが [Database ID] と [Process ID] であるビューでも有効です）。
2.	これらのデータ項目は、Adaptive Server バージョン 11.5 以降をモニタしている場合のみ使用できます。
3.	[Process ID] と [Application Name] を同じビューで使用することはできません。
4.	11.5 以前のバージョンでは、[Cache Name] はキーでした。現在、「Cache Name」は結果のデータ項目です。キー [Cache Name] は、新しいキー [Cache ID] に置き換えられています。互換性を維持するため、[Cache Name] はキーとして使用できます。ただし、効率と上位互換性を考慮して、新しいビューでは [Cache ID] をキーとして使用してください。

## 有効な組み合わせの例

- Login Name、Process ID、Page I/O
- Process ID、CPU Percent
- Login Name、Process ID
- Deadlock Count、CPU Busy Percent
- Transactions、Rows Deleted、Rows Inserted、Rows Updated
- Current Engine、Login Name、Process ID
- Current Engine、Login Name、Process ID、CPU Percent
- Login Name、Process ID、Page I/O、Connect Time

## 無効な組み合わせの例

- Current Engine、Process ID、Lock Status Count  
[Lock Status Count] はプロセス単位では使用できません。
- Login Name、Blocking Process ID  
他の必須キー・データ項目を追加する必要があります。
- Net I/O Bytes、Demand Lock  
[Demand Lock] にはキーが必要です。[Net I/O Bytes] は、どのキー・データ項目との組み合わせも無効です。
- Current Engine、Login Name、Deadlock Count  
[Deadlock Count] は、どのキー・データ項目とも組み合わせて使用することはできません。また、[Current Engine] と [Login Name] にはキーが必要です。

## 各データ項目に対して有効な統計タイプの表

6つの有効な統計タイプがあります。

- “Value for Sample”
- “Value for Session”
- “Rate for Sample”
- “Rate for Session”
- “Avg for Sample”
- “Avg for Session”

ただし、すべてのデータ項目に6つの統計タイプを使用できるわけではありません。表B-3に、各データ項目で有効な統計タイプを示します。次に表の見方を説明します。

- 空欄は、その統計タイプがそのデータ項目には無効であることを示しています。
- [有効な統計タイプ]には、その統計タイプを指定した結果、生成されるデータのデータ型を示しています。
  - long と ENUMS は整数であることを示しています。
  - double は浮動小数点数であることを示しています。
  - charp は文字列であることを示しています。

表 B-3: データ項目と有効な統計タイプ

統計タイプ データ項目	Value for Sample	Value for Session	Rate for Sample	Rate for Session	Avg for Sample	Avg for Session
Application Execution Class	charp					
Application Name	charp					
Blocking Process ID	long					
Cache Efficiency	double	double				
Cache Hit Pct	double	double				
Cache Hits	long	long	double	double		
Cache ID	long					
Cache Misses	long	long	double	double		
Cache Name	charp					
Cache Prefetch Efficiency	double	double				

各データ項目に対して有効な統計タイプの表

統計タイプ データ項目	Value for Sample	Value for Session	Rate for Sample	Rate for Session	Avg for Sample	Avg for Session
Cache Refer and Reuse	long	long	double	double		
Cache Reuse	long	long	double	double		
Cache Reuse Dirty	long	long	double	double		
Cache Size	double					
Cache Spinlock Contention	double	double				
Code Memory Size	long					
Connect Time	long	long				
CPU Busy Percent	double	double				
CPU Percent	double	double				
CPU Time	double	double				
CPU Yields	long	long	double	double		
Current Application Name	charp					
Current Engine	long					
Current Execution Class	charp					
Current Process State	ENUMS					
Current Stmt Batch ID	long					
Current Stmt Batch Text	charp					
Current Stmt Batch Text Byte Offset	long					
Current Stmt Batch Text Enabled	long					
Current Stmt Context ID	long					
Current Stmt CPU Time	double	double				
Current Stmt Elapsed Time	double	double				

統計タイプ データ項目	Value for Sample	Value for Session	Rate for Sample	Rate for Session	Avg for Sample	Avg for Session
Current Stmt Line Number	long					
Current Stmt Locks Granted Immediately	long	long	double	double		
Current Stmt Locks Granted After Wait	long	long	double	double		
Current Stmt Locks Not Granted	long	long	double	double		
Current Stmt Logical Reads	long	long	double	double		
Current Stmt Number	long					
Current Stmt Page I/O	long	long	double	double		
Current Stmt Page Writes	long	long	double	double		
Current Stmt Physical Reads	long	long	double	double		
Current Stmt Procedure Database ID	long					
Current Stmt Procedure Database Name	charp					
Current Stmt Procedure ID	long					
Current Stmt Procedure Name	charp					
Current Stmt Procedure Owner Name	charp					
Current Stmt Procedure Text	charp					
Current Stmt Query Plan Text	charp					
Current Stmt Start Time	charp					

各データ項目に対して有効な統計タイプの表

統計タイプ データ項目	Value for Sample	Value for Session	Rate for Sample	Rate for Session	Avg for Sample	Avg for Session
Current Stmt Text Byte Offset	long					
Database ID	long					
Database Name	charp					
Deadlock Count	long	long				
Demand Lock	charp					
Device Hit Percent	double	double				
Device Hits	long	long	double	double		
Device I/O	long	long	double	double		
Device Misses	long	long	double	double		
Device Name	charp					
Device Reads	long	long	double	double		
Device Writes	long	long	double	double		
Elapsed Time	long	long				
Engine Number	long					
Host Name	charp					
Index Logical Reads	long	long	double	double		
Index Physical Reads	long	long	double	double		
Kernel Process ID	long					
Kernel Structures Memory Size	long					
Large I/O Denied	long	long	double	double		
Large I/O Performed	long	long	double	double		
Large I/O Requested	long	long	double	double		
Lock Count	long	long	double	double		
Lock Hit Percent	double	double				
Lock Result	ENUMS					
Lock Results Summarized	ENUMS					

統計タイプ データ項目	Value for Sample	Value for Session	Rate for Sample	Rate for Session	Avg for Sample	Avg for Session
Lock Status	ENUMS					
Lock Status Count	long	long	double	double		
Lock Type	ENUMS					
Locks Being Blocked Count	long					
Locks Granted After Wait	long	long	double	double		
Locks Granted Immediately	long	long	double	double		
Locks Not Granted	long	long	double	double		
Log Contention Percent	double	double				
Logical Page Reads	long	long	double	double		
Login Name	charp					
Most Active Device I/O	long	long	double	double		
Most Active Device Name	charp					
Net Bytes Received	long	long	double	double		
Net Bytes Sent	long	long	double	double		
Net Default Packet Size	long					
Net I/O Bytes	long	long	double	double		
Net Max Packet Size	long					
Net Packet Size Received	double	double	double	double		
Net Packet Size Sent	double	double	double	double		
Net Packets Received	long	long	double	double		
Net Packets Sent	long	long	double	double		
Number of Engines	long					
Number of Processes	long					
Object ID	long					

各データ項目に対して有効な統計タイプの表

統計タイプ データ項目	Value for Sample	Value for Session	Rate for Sample	Rate for Session	Avg for Sample	Avg for Session
Object Name	charp					
Object Type	ENUMS					
Owner Name	charp					
Page Cache Size	long					
Lock Hit Percent	double	double				
Page I/O	long	long	double	double		
Page Number	long					
Page Writes	long	long	double	double		
Physical Page Reads	long	long	double	double		
Procedure Buffer Size	long					
Procedure CPU Time	double	double			double	double
Procedure Database ID	long					
Procedure Database Name	charp					
Procedure Elapsed Time	double	double			double	double
Procedure Execution Class	charp					
Procedure Execution Class	long	long	double	double		
Procedure Header Size	long					
Procedure Hit Percent	double	double				
Procedure ID	long					
Procedure Line Number	long					
Procedure Line Text	charp					
Procedure Logical Reads	long	long	double	double		
Procedure Name	charp					

統計タイプ データ項目	Value for Sample	Value for Session	Rate for Sample	Rate for Session	Avg for Sample	Avg for Session
Procedure Owner Name	charp					
Procedure Physical Reads	long	long	double	double		
Procedure Statement Number	long					
Process ID	long					
Process State	ENUMS					
Process State Count	long					
Rows Deleted	long	long	double	double		
Rows Deleted Deferred	long	long	double	double		
Rows Deleted Direct	long	long	double	double		
Rows Inserted	long	long	double	double		
Rows Inserted Clustered	long	long	double	double		
Rows Inserted Heap	long	long	double	double		
Rows Updated	long	long	double	double		
Rows Updated Deferred	long	long	double	double		
Rows Updated Direct	long	long	double	double		
Rows Updated Expensive	long	long	double	double		
Rows Updated In Place	long	long	double	double		
Rows Updated Not In Place	long	long	double	double		
Select Statements	long	long	double	double		
Server Structures Size	long					
SQL Server Name	charp					
SQL Server Version	charp					
Thread Exceeded Max	long	long	double	double		

各データ項目に対して有効な統計タイプの表

---

統計タイプ データ項目	Value for Sample	Value for Session	Rate for Sample	Rate for Session	Avg for Sample	Avg for Session
Thread Exceeded Max Percent	double	double				
Thread Max Used	long					
Time Waited on Lock	long					
Timestamp	charp					
Timestamp Datim	datim					
Transactions	long	long	double	double		

この付録では、プレイバック・ビューの設計規則と考慮事項について説明します。

トピック名	ページ
要約レベルの詳細	153
プレイバック・ビューの設計	157
プレイバック・ビューのデータ項目要件の表	160
データ項目に関する追加情報	165

## 要約レベルの詳細

セッションのすべてのプレイバック・ビューは、同じ要約レベルでプレイバックされます。`hs_create_playback_session` コマンドの `summarization_level` パラメータを使用して、プレイバックの要約レベルを定義します。要約レベルは次のとおりです。

- `raw`
- `actual`
- `entire`
- ユーザ定義インターバル

### raw プレイバック

このオプションでは、同じサンプル・インターバルを使用して、収集されたとおりにデータをプレイバックします。ロー・データを記録されたとおりに表示する場合は、このオプションを選択します。これは、現在の SQL 文データやプロセスごとのステータス・データなどのスナップショット・データのプレイバックに使用できる唯一のオプションです。スナップショット・データ項目の一覧については、[表 C-2 \(160 ページ\)](#) を参照してください(スナップショット・データ項目は、「`raw` 以外に使用可」カラムが「いいえ」の項目です)。

このオプションは、`target` が `client` の場合のみ有効です。

Historical Server では、データを処理しません。各プレイバック・サンプルの時刻は、入力セッションからのサンプルの時刻と完全に一致します。データは、入力セッションのものとまったく同じです。

raw プレイバックには、プレイバックの *start\_time* と *end\_time* の間に完全に収まるレコーディング・セッション・サンプルだけが含まれます。たとえば、プレイバック・セッションの *start\_time* が午後 3 時で、入力レコーディング・セッションの 1 つが午後 2 時 55 分に始まり、10 分間サンプルを取る場合、プレイバックの最初のサンプルは、午後 3 時 5 分から始まるサンプルになります。

プレイバックの *start\_time* が入力レコーディング・セッションの開始時刻より遅い場合、累積セッション値は収集されたとおりにプレイバックされ、調整は行われません。たとえば、入力レコーディング・セッションが午後 1 時に始まり、プレイバックの *start\_time* が午後 3 時の場合、データ項目 “Device I/O” “Total for Session” には、午後 1 時以降の I/O 回数が反映されます。

## actual プレイバック

このオプションでは、入力レコーディング・セッションと同じサンプル・インターバルを使用して、データをプレイバックします。プレイバック・ビューでは、記録されたデータとは異なる統計タイプを使用でき、元のビューにはない推定または計算データ項目を指定できます。プロセスやロックのステータス・データ、現在の SQL 文データなどのスナップショット・データは使用できません。プレイバック・ビューに変換または追加できるデータ項目のリストについては、[表 C-2 \(160 ページ\)](#) を参照してください。

要約の必要がなく、特定のデータ項目を追加または変更する場合は、このオプションを選択します。

このオプションは、*target* が *client* の場合のみ有効です。

Historical Server からは、時間間隔が入力セッションのサンプルの時間間隔と一致するサンプルが返されます。ただし、入力セッション間で重複が発生した場合、サンプルは指定した *start\_time* または *end\_time* に合わせてトランケートされることがあります。

要求されたプレイバックの開始時刻または終了時刻が、サンプル・インターバルの途中にある場合、Historical Server はカウントされた値を比例配分し、パーセントと比率の加重平均を計算します。重みは、各入力サンプルにかかった時間を秒数で表したものです。

たとえば、プレイバック・セッションの *start\_time* が午後 3 時で、入力レコーディング・セッションの 1 つが午後 2 時 55 分に始まり、10 分間サンプルを取る場合、プレイバックの最初のサンプルは、午後 2 時 55 分から始まるサンプルとなり、すべての値はサンプル時間の半分を反映するよう 50% の率で比例配分されます。

プレイバックの *start\_time* が入力レコーディング・セッションの開始時刻より遅い場合、累積セッション値はセッションの一部を反映するように比例配分してプレイバックされます。たとえば、入力レコーディング・セッションが午後 1 時に始まり、プレイバックの *start\_time* が午後 3 時の場合、データ項目 “Device I/O” “Total for Session” には、午後 3 時以降の I/O 回数の調整値が反映されます。

## entire プレイバック

このオプションでは、すべての入力レコーディング・セッションのデータを1つのサンプルとしてまとめてプレイバックします。たとえば、7つの入力レコード・セッションを1つのプレイバック・サンプルにまとめます。サンプル間隔は、要求されたプレイバックの *start\_time* から *end\_time* までの時間となります。

プレイバック・ビューでは、記録されたデータとは異なる統計タイプを使用でき、元のビューにはない推定または計算データ項目を指定できます。プロセスやロックのステータス・データ、現在の SQL 文データなどのスナップショット・データは使用できません。プレイバック・ビューに変換または追加できるデータ項目のリストについては、[表 C-2 \(160 ページ\)](#) を参照してください。

記録したデータを統合して、長期間にわたるアクティビティの詳細を概要情報にまとめる場合は、このオプションを選択します。

要求されたプレイバックの開始時刻または終了時刻が、入力レコーディング・セッションの途中にある場合、Historical Server はカウントされた値を比例配分し、パーセントと比率の加重平均を計算します。重みは、各入力サンプルにかかった時間を秒数で表したものです。

たとえば、プレイバック・セッションの *start\_time* が午後3時、*end\_time* が午後6時で、入力レコーディング・セッションが午後1時に開始し、午後9時に終了する場合、Historical Server は、午後3時から午後6時までの時間のデータ項目値を計算します。プレイバックに指定した開始時刻と終了時刻が、入力セッションの開始時刻および終了時刻と一致しない場合、累積(セッション)は比例配分され、累積パーセントと比率には加重平均が使用されます。

## ユーザ定義の間隔でのプレイバック

このオプションでは、指定した長さのサンプル・インターバルに要約されたデータをプレイバックします。パラメータ値はサンプル・インターバルで、次のように指定します。

“S”  
“M:S”  
“H:M:S”  
“D H:M:S”

パラメータの意味は次のとおりです。

- *S* は秒
- *M* は分
- *H* は時間
- *D* は日

すべてのコンポーネントは 1 桁または 2 桁の数値です。以下は、その例です。

```
"30" (specifies sample intervals of 30 seconds)
"10:0" (specifies sample intervals of 10 minutes) "8:30:0"
(specifies sample intervals of 8 1/2 hours)
"5 0:0:0" (specifies sample intervals of 5 days)
```

最初のサンプル間隔は *start\_time* に開始し、最後のサンプルを除くすべてのサンプルが指定された長さを持ちます。要求されたプレイバックの開始時刻または終了時刻が、元のレコーディング・セッションのサンプル間隔の途中にある場合、Historical Server はプレイバック・サンプルの記録済みデータ値を比例配分します。最後のサンプルは、必要に応じて、指定した *end\_time* まで短縮されることがあります。

データを任意の細分性で要約する場合は、このオプションを選択します。この要約タイプでは、アクティビティの偏りを調整できるので、長期的な傾向を調べるときに便利です。

プレイバック・ビューでは、記録されたデータとは異なる統計タイプを使用でき、元のビューにはない推定または計算データ項目を指定できます。プレイバック・ビューに変換または追加できるデータ項目のリストについては、[表 C-2 \(160 ページ\)](#) を参照してください。

Historical Server は、サンプル・インターバルごとにカウントされた値を比例配分し、パーセントと比率に加重平均を計算して、要求されたプレイバック・サンプル・インターバルのデータを調整します。重みは、各入力サンプルにかかる時間を秒数で表したものです。

## 要約インターバルのまとめ

表 C-1 は、各 *summarization\_interval* 値のプレイバック機能を示します。

表 C-1: 要約インターバルの機能

機能	raw	actual	entire	ユーザ定義の間隔
プレイバックにはすべての記録済みデータ項目を使用できる	はい	いいえ	いいえ	いいえ
プレイバック・ビューでタイムスタンプを使用できる	はい	はい	はい	はい
プレイバック・ビューで統計タイプを変更できる	いいえ	はい	はい	はい
ファイルへのプレイバック(要約レコーディング・セッションの作成)ができる	いいえ	いいえ	はい	はい
入力ビューにない計算データ項目または推定データ項目をプレイバック・ビューで使用できる	いいえ	はい	はい	はい

機能	raw	actual	entire	ユーザ定義の間隔
プレイバックが入力セッションの途中から始まる場合、セッション(累積)データが比例配分される	いいえ	はい	はい	はい
プレイバックが入力セッションの途中から始まる場合、サンプルは省略される	いいえ	はい	はい	はい
要約	いいえ	いいえ	はい	はい
適切なデータ調整による標準化されたサンプル・インターバル	いいえ	いいえ	いいえ	はい

## プレイバック・ビューの設計

この項では、次の項目について説明します。

- [入力セッションの指定規則](#)
- [入力ビューとプレイバック・ビューの関係](#)
- [raw プレイバック・ビューの定義規則](#)
- [raw 以外のプレイバック・ビューの定義規則](#)

### 入力セッションの指定規則

複数の入力セッションを使用してファイルにプレイバック・セッションを作成する場合は、入力セッション間に空白の時間が発生しないようにしてください。つまり、*target* が *file* の場合、入力セッションの終了時刻と次の入力セッションの開始時刻が同じになるように指定する必要があります。

たとえば、月曜日から金曜日まで毎日、午前 9 時から午後 5 時までデータを収集する場合、これらの 5 つのレコーディング・セッションをプレイバックして、新しく要約された週のセッションを作成することはできません。ただし、月曜日から金曜日まで毎日、午前 9 時から午前 9 時までデータを収集して空白時間を解消すれば、プレイバック機能を使用して新しく要約された週のセッションを作成できます。午前 9 時から午後 5 時までのレコーディング・セッションを維持し、これとは別に午後 5 時から午前 9 時までのレコーディング・セッションを追加して、空白時間を解消する方法もあります。オフ時間のサンプル間隔を長くすると、収集するデータの量が減ります。

## 入力ビューとプレイバック・ビューの関係

`hs_create_playback_session` コマンドを使用して、プレイバック・セッションを作成します。次に、1つ以上の `hs_create_playback_view` コマンドを使用して、プレイバック・セッションにビューを追加します。プレイバック・セッションには、少なくとも1つのビューが必要です。プレイバック・セッションには、複数のビューを指定できます。

`hs_create_playback_session` コマンドで、プレイバックに含める入力セッション ID を指定します。入力ビューは、入力レコーディング・セッションに定義されたビューです。プレイバック・ビューでは、プレイバックする対応する入力ビューのデータ項目を定義します。

プレイバック・ビューは、入力ビューから導出されます。プレイバック・ビューの名前は、入力セッションのビュー名と一致している必要があります。複数の入力セッションを使用する場合は、ビューがすべての入力セッションに存在し、すべての入力セッションのビュー定義に同じデータ項目リストと同じフィルタ・リストが指定されている必要があります。

## ビューの定義規則

プレイバック・ビューの定義規則は、プレイバック・セッションに定義された *summarization\_interval* が `raw` または `raw` 以外のどちらに設定されているかによって異なります。

### raw プレイバック・ビューの定義規則

プレイバック・セッションの *summarization\_interval* が `raw` の場合は、次の規則に従います。

- プレイバック・ビューには、入力ビューのすべてのキー・データ項目を必ず指定します。[表 C-2](#) の「キー」というラベルの付いたカラムは、キー・データ項目を示します。キーに有効な統計タイプは、“Value for Sample”だけです。
- プレイバック・ビューでは、入力ビューの結果データ項目はオプションです。
- プレイバック・ビューに表示するすべてのデータ項目の統計タイプは、入力ビューと同じタイプである必要があります。
- プレイバック・ビューでは、次のデータ項目を使用できます。

“Timestamp” “Value for Sample”

“Timestamp Datim” “Value for Sample”

“Elapsed Time” “Value for Sample”

“Elapsed Time” “Value for Session”

入力ビューのすべてのデータ項目をプレイバックし、タイムスタンプを追加する必要がない場合は、*data\_item\_name\_n* パラメータと *data\_item\_stat\_n* パラメータを使用しないで *hs\_create\_playback\_view* コマンドを実行します。これらのパラメータを省略すると、デフォルトでプレイバック・ビューは同じ名前の入力ビューからすべてのデータ項目を使用して定義されます。

## raw 以外のプレイバック・ビューの定義規則

*summarization\_interval* が *entire*、*actual*、または要約インターバルの場合は、次の規則に従います。

- プレイバック・ビューには、入力ビューのすべてのキー・データ項目を必ず指定します。[表 C-2 \(160 ページ\)](#) の「キー」というラベルの付いたカラムは、キー・データ項目を示します。キーに有効な統計タイプは、“Value for Sample”だけです。
- プレイバック・ビューでは、次のデータ項目を使用できます。  
“Timestamp” “Value for Sample”  
“Timestamp Datim” “Value for Sample”  
“Elapsed Time” “Value for Sample”  
“Elapsed Time” “Value for Session”
- [表 C-2](#) の「raw 以外に使用可」というラベルの付いたカラムは、raw 以外のプレイバック・ビューで使用できる結果データ項目を示します。「指定条件」というラベルの付いたカラムは、データ項目を使用するための条件を表します。
- 「指定条件」が「フル」の場合は、入力ビューで使用されている統計タイプに関係なく、そのデータ項目に有効なすべての統計タイプを使用できます。各データ項目に有効な統計タイプについては、[表 B-3 \(145 ページ\)](#) を参照してください。
- 「指定条件」カラムが「推定」の場合、プレイバック・ビューにデータ項目を指定するには、次の両方の条件を満たしている必要があります。
  - 入力ビューのデータ項目の統計タイプが “Value for Sample” であり、なおかつ
  - プレイバック・セッションが、推定を許可するように定義されている (*hs\_create\_playback\_session* コマンドで、*allow\_estimation* パラメータが *allow* に設定されている)
 入力ビューで使用されている統計タイプに関係なく、プレイバック・ビューでは、そのデータ項目に有効なすべての統計タイプを使用できます。
- 「指定条件」カラムが「計算」の場合、プレイバック・ビューにデータ項目を指定するには、次のどちらかの条件を満たしている必要があります。

- 入力ビューに、「次のデータ項目から計算」というラベルの付いたカラムにリストされたデータ項目が存在すること。存在する場合は、プレイバック中に Historical Server によって値が計算されます。または
- 計算に必要なデータ項目が入力ビューにない場合は、Historical Server によってプレイバック中に値が推定されること。この場合、前に説明した推定の条件が満たされている必要があります。

入力ビューで使用されている統計タイプに関係なく、プレイバック・ビューでは、そのデータ項目に有効なすべての統計タイプを使用できます。

## プレイバック・ビューのデータ項目要件の表

表 C-2 は、プレイバック・ビューのデータ項目を指定するための要件を示します。各カラムの説明については、前の項の説明をお読みください。

表 C-2: プレイバック・ビューのデータ項目要件

データ項目 (プレイバック・ビューには入力ビューのキーが必要)	raw 以外に使用可	指定条件
Application Execution Class	有効	Value for Sample
Application Name (キー)		いいえ
Blocking Process ID		有効
Cache Efficiency		推定
Cache Hit Pct	有効	次のデータ項目から計算 • Cache Hits • Cache Misses
Cache Hits	有効	フル
Cache ID (キー) <sup>1</sup>		
Cache Misses	有効	フル
Cache Name <sup>1</sup>	有効	Value for Sample
Cache Prefetch Efficiency	有効	推定
Cache Refer And Reuse	有効	フル
Cache Reuse	有効	フル
Cache Reuse Dirty	有効	フル
Cache Size	有効	Value for Sample
Cache Spinlock Contention	有効	推定
Code Memory Size	有効	Value for Sample
Connect Time	有効	フル
CPU Busy Percent	有効	推定
CPU Percent	有効	推定
CPU Time	有効	フル

データ項目 (プレイバック・ビューには入力 ビューのキーが必要)	raw 以外に使 用可	指定条件
CPU Yields	有効	
Current Application Name	いいえ	
Current Engine	いいえ	
Current Execution Class	いいえ	
Current Process State	いいえ	
Current Stmt Batch ID	いいえ	
Current Stmt Batch Text	いいえ	
Current Stmt Batch Text Byte Offset	いいえ	
Current Stmt Batch Text Enabled	いいえ	
Current Stmt Cache Reads	いいえ	
Current Stmt Context ID	いいえ	
Current Stmt CPU Time	いいえ	
Current Stmt Elapsed Time	いいえ	
Current Stmt Line Number	いいえ	
Current Stmt Locks Granted Immediately	いいえ	
Current Stmt Locks Granted After Wait	いいえ	
Current Stmt Locks Not Granted	いいえ	
Current Stmt Max Text Config Size	いいえ	
Current Stmt Max Text Default Size	いいえ	
Current Stmt Max Text Run Size	いいえ	
Current Stmt Number	いいえ	
Current Stmt Page I/O	いいえ	
Current Stmt Page Writes	いいえ	
Current Stmt Physical Reads	いいえ	
Current Stmt Procedure Database ID	いいえ	
Current Stmt Procedure Database Name	いいえ	
Current Stmt Procedure ID	いいえ	
Current Stmt Procedure Name	いいえ	
Current Stmt Procedure Owner Name	いいえ	
Current Stmt Procedure Text	いいえ	
Current Stmt Query Plan Text	いいえ	
Current Stmt Start Time	いいえ	
Current Stmt Text Byte Offset	いいえ	
Database ID (キー)	有効	
Database Name	有効	Value for Sample
Deadlock Count	有効	フル
Demand Lock	いいえ	

プレイバック・ビューのデータ項目要件の表

データ項目 (プレイバック・ビューには入力 ビューのキーが必要)	raw 以外に使 用可	指定条件
Device Hit Percent	有効	次のデータ項目から 計算 • Device Hits • Device Misses
Device Hits	有効	フル
Device I/O	有効	フル
Device Misses	有効	フル
Device Name (キー)	有効	Value for Sample
Device Reads	有効	フル
Device Writes	有効	フル
Elapsed Time <sup>2</sup>	有効	フル
Engine Number (キー)	有効	Value for Sample
Host Name	いいえ	
Index Logical Reads	有効	フル
Index Physical Reads	有効	フル
Kernel Process ID (キー)	有効	Value for Sample
Kernel Structures Memory Size	有効	Value for Sample
Large I/O Denied	有効	フル
Large I/O Performed	有効	フル
Large I/O Requested	有効	フル
Lock Count	有効	フル
Lock Hit Percent	有効	推定
Lock Result (キー)	有効	Value for Sample
Lock Results Summarized (キー)	有効	Value for Sample
Lock Status (キー)	有効	Value for Sample
Lock Status Count	いいえ	
Lock Type (キー)	有効	Value for Sample
Locks Being Blocked Count	いいえ	
Locks Granted Immediately	有効	フル
Locks Granted After Wait	有効	フル
Locks Not Granted	有効	フル
Log Contention Percent	有効	推定
Logical Page Reads	有効	フル
Login Name	有効	Value for Sample
Most Active Device I/O	いいえ	
Most Active Device Name	いいえ	
Net Bytes Received	有効	フル
Net Bytes Sent	有効	フル
Net Default Packet Size	有効	Value for Sample

データ項目(プレイバック・ビューには入力 ビューのキーが必要)	raw以外に使 用可	指定条件
Net I/O Bytes	有効	フル
Net Max Packet Size	有効	Value for Sample
Net Packet Size Received	有効	次のデータ項目から 計算 • Net Packets Received • Net Bytes Received
Net Packet Size Sent	有効	次のデータ項目から 計算 • Net Packets Sent • Net Bytes Sent
Net Packets Received	有効	フル
Net Packets Sent	有効	フル
Number of Engines	有効	Value for Sample
Number of Processes	有効	推定
Object ID(キー)	有効	Value for Sample
Object Name	有効	Value for Sample
Object Type	有効	Value for Sample
Owner Name	有効	Value for Sample
Page Cache Size	有効	Value for Sample
Lock Hit Percent	有効	次のデータ項目から 計算 • Logical Reads • Physical Reads
Page I/O	有効	フル
Page Number(キー)	有効	Value for Sample
Page Writes	有効	フル
Physical Page Reads	有効	フル
Procedure Buffer Size	有効	Value for Sample
Procedure CPU Time		
Procedure Database ID(キー)	有効	Value for Sample
Procedure Database Name	有効	フル
Procedure Elapsed Time	有効	必要なデータ項目： Procedure Execution Count 詳細については、この マニュアルの 「Procedure Elapsed Time と Procedure CPU Time の使用」を参照。
Procedure Execution Class	いいえ	

プレイバック・ビューのデータ項目要件の表

データ項目 (プレイバック・ビューには入力 ビューのキーが必要)	raw 以外に使 用可	指定条件
Procedure Execution Class	有効	フル
Procedure Header Size	有効	Value for Sample
Procedure Hit Percent	有効	次のデータ項目から 計算 • Procedure Logical Reads • Procedure Physical Reads
Procedure ID (キー)	有効	Value for Sample
Procedure Line Number (キー)	有効	Value for Sample
Procedure Line Text	有効	Value for Sample
Procedure Logical Reads	有効	フル
Procedure Name	有効	Value for Sample
Procedure Owner Name	有効	Value for Sample
Procedure Physical Reads	有効	フル
Procedure Statement Number (キー)	有効	Value for Sample
Process ID (キー)	有効	Value for Sample
Process State (キー)	有効	Value for Sample
Process State Count	いいえ	
Rows Deleted	有効	フル
Rows Deleted Deferred	有効	フル
Rows Deleted Direct	有効	フル
Rows Inserted	有効	フル
Rows Inserted Clustered	有効	フル
Rows Inserted Heap	有効	フル
Rows Updated	有効	フル
Rows Updated Deferred	有効	フル
Rows Updated Direct	有効	フル
Rows Updated Expensively	有効	フル
Rows Updated In Place	有効	フル
Rows Updated Not In Place	有効	フル
Select Statements	有効	フル
Server Structures Size	有効	Value for Sample
SQL Server Name	有効	Value for Sample
SQL Server Version	有効	Value for Sample
Thread Exceeded Max	有効	フル
Thread Exceeded Max Percent	有効	フル
Thread Max Used	いいえ	
Time Waited on Lock	いいえ	

データ項目 (プレイバック・ビューには入力ビューのキーが必要)	raw 以外に使用可	指定条件
Timestamp <sup>2</sup>	有効	Value for Sample
Timestamp Datim <sup>2</sup>	有効	Value for Sample
Transactions	有効	フル
1. 11.5 以前のバージョンでは、[Cache Name] はキーでした。現在、「Cache Name」は結果のデータ項目です。キー [Cache Name] は、新しいキー [Cache ID] に置き換えられています。互換性を維持するため、[Cache Name] はキーとして使用できます。ただし、効率と上位互換性を考慮して、新しいビューでは [Cache ID] をキーとして使用してください。		
2. これらのデータ項目は、入力ビューになくてもプレイバック・ビューに表示できます。		

## データ項目に関する追加情報

この項では、データ項目をビューで使用する場合の追加情報について説明します。

### “Timestamp”、“Timestamp Datim”、“Elapsed Time” の使用

次のデータ項目は、入力ビューになくても常にプレイバック・ビューに表示できます。

“Timestamp” “Value for Sample”  
 “Timestamp Datim” “Value for Sample”  
 “Elapsed Time” “Value for Sample”  
 “Elapsed Time” “Value for Session”

これらの 1 つ以上の時間関連データ項目だけで構成されるプレイバック・ビューを作成できます。このようなビューを作成するには、入力セッションの有効なビュー名が必要です。入力セッションには、プレイバックする予定のないビューも含まれています。

*summarization\_level* が **raw** の場合、“Timestamp” と “Timestamp Datim” の値は元のデータ・ファイルと同じで、最初の記録時に Adaptive Server によって報告された時間です。*summarization\_level* が **actual**、**entire** またはユーザ定義インターバルの場合、“Timestamp” と “Timestamp Datim” の値には、要約を行った Historical Server システムのタイム・ゾーンが反映されます。

## プロセス ID の使用

サーバ・プロセスが終了すると、Adaptive Server はそのプロセス ID を新しいプロセスのために再使用することができます。したがって、[Process ID] データ項目では、プロセスをユニークに識別できるとは限りません。ただし、[Kernel Process ID] データ項目を使用すると、プロセスをユニークに識別できます。

プロセス単位のデータを表示する raw 以外のプレイバック・ビューを作成するには、入力ビューとプレイバック・ビューに、[Process ID] と [Kernel Process ID] の両方のデータ項目が必要です。プレイバックが raw の場合、使用できるのは [Process ID] だけです。

Process ID を含むビューは、次のようにローを返します。

- レコーディング・セッション・ビューと raw プレイバック・ビューの場合  
これらのビューは、サンプル間隔の最後に存在したプロセス ID のローだけを返します。サンプル間隔の途中でサーバ・プロセスが終了した場合、そのプロセス ID のローは返されません。
- セッションの *summarization\_level* パラメータが *actual*、*entire*、またはユーザ定義の間隔であるプレイバック・ビューの場合  
これらのビューは、入力ビューを構成するすべてのサーバ・プロセスのローを返します。ただし、プロセス ID がユニークであるという保証はないため、キーの一意性を保証するには [Kernel Process ID] データ項目を使用します。このデータ項目がないとエラーとなり、2 つのプロセスが別々に要約されることがあります。

## Procedure Elapsed Time と Procedure CPU Time の使用

この項では、“Avg for Sample” と “Avg for Session” 統計タイプを使用できるデータ項目の使用方法を説明します。データ項目は次のとおりです。

- Procedure Elapsed Time
- Procedure CPU Time

プレイバック・ビューで、統計タイプを “Value for Sample” または “Value for Session” に設定してこれらのデータ項目名を使用する場合、入力ビューでも同じ統計タイプを使用する必要があります。

プレイバック・ビューで、統計タイプを “Avg for Sample” または “Avg for Session” に設定してこれらのデータ項目名を使用すると、次の処理が実行されます。

- 入力ビューで追加データ項目として Procedure Execution Count が指定されている場合、戻り値が計算されます。

- 次のすべてに該当する場合、戻り値は推定されます。
  - 入力ビューに Procedure Execution Count がない
  - 入力ビューに、統計タイプが “Avg for Sample” の Procedure Elapsed Time が存在する
  - プレイバック・セッションで推定が許可されている。つまり、`hs_create_playback_session` コマンドで、`allow_estimation` パラメータが `allow` に設定されている。



## レコーディング・セッション・ビューの例

この付録では、Historical Server の有効なビューの例を紹介します。これらのビューは、インストール・ディレクトリの *sample/histserver* サブディレクトリにインストールされた *views* ファイルにも収められています。

そのまま使用して必要な情報を適切に収集できるビューもあれば、必要なビューを作成するためのテンプレートとして機能するビューもあります。

サンプル・ビューには、データを蓄積する時間間隔(レコーディング・セッションの継続時間または最新のサンプル・インターバルのみ)だけが他と異なるものがあります。同様のデータ項目を異なる順序で表示するビューもあります。データはキー・フィールドに従ってソートされるので、ビュー内に表示されるデータ項目の順序は重要です。ビュー定義の先頭にあるキー・フィールドはプライマリ・ソート・キー、2番目のキー・フィールドはセカンダリ・ソート・キーとして動作する、というように以降も続きます。

次の理由から、*views* ファイルを *isql* への入力として直接使用しないでください。

- 1つのレコーディング・セッションで、このファイルに含まれているすべてのビューを定義することはほとんどありません。
- ファイル内には、ビュー定義の実際のテキストを含む行とコメント行が混在しています。Historical Server では、コメント行は認識されません。ビュー定義をカット・アンド・ペーストするときは、コメント行を無視してください。

トピック名	ページ
キャッシュ・パフォーマンスの概要	170
データベース・オブジェクトのロック・ステータス	170
データベース・オブジェクトのページ I/O	171
個別キャッシュに対するデータ・キャッシュ・アクティビティ	172
レコーディング・セッションにおけるデータ・キャッシュ統計	172
サンプル・インターバルにおけるデータ・キャッシュ統計	173
レコーディング・セッションにおけるデバイス I/O	173
サンプル間隔におけるデバイス I/O	174
デバイス I/O パフォーマンスの概要	174
エンジン・アクティビティ	175
ロック・パフォーマンスの概要	175
レコーディング・セッション中のネットワーク・アクティビティ	175

トピック名	ページ
サンプル・インターバル中のネットワーク・アクティビティ	176
ネットワーク・パフォーマンスの概要	176
ページ I/O	177
レコーディング・セッションにおけるプロシージャ・キャッシュ統計	177
サンプル・インターバルにおけるプロシージャ・キャッシュ統計	178
プロシージャ・ページ I/O	178
プロセス・アクティビティ	179
プロセスのデータベース・オブジェクト・ページ I/O	179
プロセスのロック詳細情報	180
プロセスのページ I/O 詳細情報	180
プロセスのロック情報	181
プロセスのページ I/O 情報	181
プロセス・ステータスの概要	182
プロセスのストアド・プロシージャ・ページ I/O	182
サーバ・パフォーマンスの概要	183
ストアド・プロシージャ・アクティビティ	183
トランザクション・アクティビティ	184

## キャッシュ・パフォーマンスの概要

このビューは、最新のサンプル・インターバル中における Adaptive Server のキャッシュの全体的な効率を示します。Adaptive Server データ・キャッシュで対応できたデータ・ページ読み込みの割合と、Adaptive Server プロシージャ・キャッシュで対応できたプロシージャ実行要求の割合が表示されます。

```
hs_create_view cache_perf_sum,
"Page Hit Percent", "Value for Sample",
"Procedure Hit Percent", "Value for Sample"
```

## データベース・オブジェクトのロック・ステータス

このビューは、最新のサンプル・インターバルが終了した時点で、Adaptive Server プロセスによって保持または要求されているデータベース・オブジェクトのロック・ステータスを示します。各ロックは、ロックされているオブジェクトの名前と ID、そのオブジェクトが属するデータベースの名前と ID、ロックを適用するページ番号 (ページ・ロックの場合) によって識別されます。

ロックと対応する Adaptive Server の各プロセスも、ログイン名、プロセス ID、カーネル・プロセス ID によって特定されます。ロックの現在のステータスおよびロックがデマンド・ロックであるかどうかを示すとともに、ロックのタイプも示されます。プロセスがロックを要求している場合は、このプロセスがロックの取得に待機した時間とロックを保持しているプロセスのプロセス ID が表示されます。プロセスがすでにロックを保持している場合は、そのロックの取得を待機中のプロセス数が表示されます。

```
hs_create_view object_lock_status,
"Database ID", "Value for Sample",
"Database Name", "Value for Sample",
"Object ID", "Value for Sample",
"Object Name", "Value for Sample",
"Page Number", "Value for Sample",
"Login Name", "Value for Sample",
"Process ID", "Value for Sample",
"Kernel Process ID", "Value for Sample",
"Lock Type", "Value for Sample",
"Lock Status", "Value for Sample",
"Demand Lock", "Value for Sample",
"Time Waited on Lock", "Value for Sample",
"Blocking Process ID", "Value for Sample",
"Locks Being Blocked Count", "Value for Sample"
```

## データベース・オブジェクトのページ I/O

このビューは、Adaptive Server データベースのオブジェクトとこれらに対応するページ I/O を示します。Adaptive Server のデータベース名と ID、各データベース内のオブジェクト名と ID が表示されます。このビューには、最新のサンプル・インターバルとレコーディング・セッションの両方における、各オブジェクトに対応する論理ページ読み込み、物理ページ読み込み、ページ書き込みが表示されます。

```
hs_create_view object_page_io,
"Database ID", "Value for Sample",
"Database Name", "Value for Sample",
"Object ID", "Value for Sample",
"Object Name", "Value for Sample",
"Logical Page Reads", "Value for Sample",
"Physical Page Reads", "Value for Sample",
"Page Writes", "Value for Sample",
"Logical Page Reads", "Value for Session",
"Physical Page Reads", "Value for Session",
"Page Writes", "Value for Session"
```

## 個別キャッシュに対するデータ・キャッシュ・アクティビティ

このビューは、SQL Server バージョン 11.0.x および Adaptive Server バージョン 11.5 以降でのみ使用できます。このビューには、個々のデータ・キャッシュのパフォーマンスに関する情報が収められます。

このビューは、Adaptive Server で設定した名前付きの各キャッシュとデフォルトのデータ・キャッシュについて、キャッシュの名前と、レコーディング・セッション開始以降にキャッシュで対応できたキャッシュにバインドされているオブジェクトのページ読み込みの割合を収集します。

このビューは、セッション開始以降のキャッシュによる領域使用率、キャッシュによるスピンドル取得の待機回数の割合を示します。セッションの実際のキャッシュ・ヒット数と未ヒット数も収集されます。

```
hs_create_view data_cache_sum,  
"Cache Name", "Value for Sample",  
"Cache Hit Pct", "Value for Session",  
"Cache Efficiency", "Value for Session",  
"Cache Spinlock Contention", "Value for Session",  
"Cache Hits", "Value for Session",  
"Cache Misses", "Value for Session"
```

## レコーディング・セッションにおけるデータ・キャッシュ統計

このビューは、レコーディング・セッション開始以降の Adaptive Server のデータ・キャッシュ全体の総合的な効率を示します。レコーディング・セッション中にキャッシュで対応できたページ読み込み要求の割合が表示されます。

レコーディング・セッションの論理ページ読み込み、物理ページ読み込み、ページ書き込みの回数と比率も表示されます。

```
hs_create_view session_page_cache_stats,  
"Page Hit Percent", "Value for Session",  
"Logical Page Reads", "Value for Session",  
"Logical Page Reads", "Rate for Session",  
"Physical Page Reads", "Value for Session",  
"Physical Page Reads", "Rate for Session",  
"Page Writes", "Value for Session",  
"Page Writes", "Rate for Session"
```

## サンプル・インターバルにおけるデータ・キャッシュ統計

このビューは、最新のサンプル・インターバルについて、Adaptive Server のデータ・キャッシュ全体の総合的な効率を示します。最新のサンプル・インターバル中にキャッシュで対応できたページ読み込み要求の割合が表示されます。

最新のサンプル・インターバルにおける論理ページ読み込み、物理ページ読み込み、ページ書き込みの数と比率も表示されます。

```
hs_create_view sample_page_cache_stats,
"Page Hit Percent", "Value for Sample",
"Logical Page Reads", "Value for Sample",
"Logical Page Reads", "Rate for Sample",
"Physical Page Reads", "Value for Sample",
"Physical Page Reads", "Rate for Sample",
"Page Writes", "Value for Sample",
"Page Writes", "Rate for Sample"
```

## レコーディング・セッションにおけるデバイス I/O

このビューは、レコーディング・セッション開始以降に Adaptive Server データベース・デバイスで発生した I/O アクティビティを示します。各デバイスは、名前で識別します。デバイス I/O のレベルは次の 2 つの方法で示されます。1 つはセッション開始からのデバイス I/O の合計回数、読み込み回数、書き込み回数を示す方法で、もう 1 つはセッション開始からの I/O 合計、読み込み、書き込みの 1 秒あたりの平均回数を示す方法です。

```
hs_create_view session_device_io,
"Device Name", "Value for Sample",
"Device Reads", "Value for Session",
"Device Writes", "Value for Session",
"Device I/O", "Value for Session",
"Device Reads", "Rate for Session",
"Device Writes", "Rate for Session",
"Device I/O", "Rate for Session"
```

## サンプル間隔におけるデバイス I/O

このビューは、最新のサンプル間隔において Adaptive Server データベース・デバイスで発生した I/O アクティビティを示します。各デバイスは、名前で識別します。デバイス I/O のレベルは次の 2 つの方法で示されます。1 つは最新のサンプル間隔におけるデバイス I/O の合計回数、読み込み回数、書き込み回数を示す方法で、もう 1 つはサンプル間隔における I/O 合計、読み込み、書き込みの 1 秒あたりの平均回数を示す方法です。

```
hs_create_view sample_device_io,
"Device Name", "Value for Sample",
"Device I/O", "Value for Sample",
"Device Reads", "Value for Sample",
"Device Writes", "Value for Sample",
"Device I/O", "Rate for Sample",
"Device Reads", "Rate for Sample",
"Device Writes", "Rate for Sample"
```

## デバイス I/O パフォーマンスの概要

このビューは、レコーディング・セッションの開始以降に累積された、Adaptive Server によるデータベース・デバイスへの読み込みと書き込みを示します。セッション開始以降に発生したデータベース・デバイスへの読み込みと書き込みの全体的な比率、その期間中に最もアクティブだったデータベース・デバイス、最もアクティブだったデバイスへの読み込みおよび書き込み比率が表示されます。

```
hs_create_view device_perf_sum,
"Device I/O", "Rate for Session",
"Most Active Device Name", "Value for Session",
"Most Active Device I/O", "Rate for Session"
```

## エンジン・アクティビティ

このビューは、アクティブな Adaptive Server エンジンごとに最新サンプル間隔の間のアクティビティのレベルを示します。エンジンごとに、そのエンジンが CPU を使用したときのサンプル間隔の割合が表示されます。

サンプル・インターバル中にそのエンジンで生成された論理ページの読み込み回数、物理ページの読み込み回数、ページ書き込み回数も表示されます。

```
hs_create_view engine_activity,
"Engine Number", "Value for Sample",
"CPU Busy Percent", "Value for Sample",
"Logical Page Reads", "Value for Sample",
"Physical Page Reads", "Value for Sample",
"Page Writes", "Value for Sample"
```

## ロック・パフォーマンスの概要

このビューは、最新のサンプル・インターバル中に要求および許可されたロックの回数をロック・タイプごとに示します。

```
hs_create_view lock_perf_sum,
"Lock Type", "Value for Sample",
"Lock Results Summarized", "Value for Sample",
"Lock Count", "Value for Sample"
```

## レコーディング・セッション中のネットワーク・アクティビティ

このビューは、レコーディング・セッション開始以降に発生した Adaptive Server ネットワーク接続全体のネットワーク・アクティビティを示します。デフォルトのパケット・サイズ、最大パケット・サイズ、セッション開始以降に送受信したパケットの平均サイズが表示されます。このビューは、送信済みパケット数、受信済みパケット数、パケット単位の送受信速度を示します。送信済みバイト数、受信済みバイト数、バイト単位の送受信速度も表示されます。

```
hs_create_view session_network_activity,
"Net Default Packet Size", "Value for Sample",
"Net Max Packet Size", "Value for Sample",
"Net Packet Size Sent", "Value for Session",
"Net Packet Size Received", "Value for Session",
"Net Packets Sent", "Value for Session",
"Net Packets Received", "Value for Session",
"Net Packets Sent", "Rate for Session",
"Net Packets Received", "Rate for Session",
"Net Bytes Sent", "Value for Session",
```

```
"Net Bytes Received", "Value for Session",
"Net Bytes Sent", "Rate for Session",
"Net Bytes Received", "Rate for Session"
```

## サンプル・インターバル中のネットワーク・アクティビティ

このビューは、最新のサンプル・インターバル中に発生した Adaptive Server ネットワーク接続全体のネットワーク・アクティビティを示します。デフォルトのパケット・サイズ、最大パケット・サイズ、サンプル・インターバル中に送受信したパケットの平均サイズが表示されます。このビューは、送信済みパケット数、受信済みパケット数、パケット単位の送受信速度を示します。送信済みバイト数、受信済みバイト数、バイト単位の送受信速度も表示されます。

```
hs_create_view sample_network_activity,
"Net Default Packet Size", "Value for Sample",
"Net Max Packet Size", "Value for Sample",
"Net Packet Size Sent", "Value for Sample",
"Net Packet Size Received", "Value for Sample",
"Net Packets Sent", "Value for Sample",
"Net Packets Received", "Value for Sample",
"Net Packets Sent", "Rate for Sample",
"Net Packets Received", "Rate for Sample",
"Net Bytes Sent", "Value for Sample",
"Net Bytes Received", "Value for Sample",
"Net Bytes Sent", "Rate for Sample",
"Net Bytes Received", "Rate for Sample"
```

## ネットワーク・パフォーマンスの概要

このビューは、最新のサンプル・インターバル中に発生したネットワーク接続全体の Adaptive Server アクティビティの比率を示します。サンプル・インターバル中に Adaptive Server によって送受信された 1 秒間当たりのバイト数が表示されます。

```
hs_create_view network_perf_sum,
"Net Bytes Received", "Rate for Sample",
"Net Bytes Sent", "Rate for Sample"
```

## ページ I/O

このビューでは、レコーディング・セッション中に最もページ読み込み回数の多かったテーブルを特定できます。ページ読み込みはサンプル単位で累積されるので、1日のうちで異なる時間帯のテーブルを対象にしたアクティビティも追跡できます。レコーディング・セッションのサンプル・インターバルを、1日のうちの異なる時間帯におけるアクティビティの傾向を測定するのに最も有効な細分性のレベルで設定できます。

レコーディング・セッション全体のページ読み込み回数だけを蓄積することもできます。この場合、サンプル・インターバルをレコーディング・セッションの長さとほぼ等しくなるように設定し、ビューから“Logical Page Reads”と“Value for Sample”を除外します。

```
hs_create_view page_ios,
"Database ID", "Value for Sample",
"Database Name", "Value for Sample",
"Object ID", "Rate for Sample",
"Logical Page Reads", "Value for Sample",
"Logical Page Reads", "Rate for Session"
```

---

**注意** このビューで特定のデータベース・テーブルに関して収集されるページ I/O 数には、テーブル自体のデータ・ページに対して実行された I/O に加えて、テーブルのインデックスに対する I/O も含まれます。

---

## レコーディング・セッションにおけるプロシージャ・キャッシュ統計

このビューは、レコーディング・セッション開始以降における Adaptive Server のプロシージャ・キャッシュの効率を示します。プロシージャ・キャッシュで対応できたストアド・プロシージャ実行要求の割合が表示されます。

ビューには、セッション開始以降のストアド・プロシージャの論理読み込み回数と物理読み込み回数、セッション開始以降のストアド・プロシージャの論理および物理読み込みの全体的な比率も表示されます。

```
hs_create_view session_procedure_cache_stats,
"Procedure Hit Percent", "Value for Session",
"Procedure Logical Reads", "Value for Session",
"Procedure Logical Reads", "Rate for Session",
"Procedure Physical Reads", "Value for Session",
"Procedure Physical Reads", "Rate for Session"
```

## サンプル・インターバルにおけるプロシージャ・キャッシュ統計

このビューは、最新のサンプル・インターバルにおける Adaptive Server のプロシージャ・キャッシュの効率を示します。最新のサンプル・インターバル中に、プロシージャ・キャッシュで対応できたストアド・プロシージャ実行要求の割合が表示されます。

ビューには、最新のサンプル・インターバル中のストアド・プロシージャの論理読み込み回数と物理読み込み回数、最新のサンプル・インターバル中のストアド・プロシージャの論理および物理読み込みの全体的な比率も表示されます。

```
hs_create_view sample_procedure_cache_stats,
"Procedure Hit Percent", "Value for Sample",
"Procedure Logical Reads", "Value for Sample",
"Procedure Logical Reads", "Rate for Sample",
"Procedure Physical Reads", "Value for Sample",
"Procedure Physical Reads", "Rate for Sample"
```

## プロシージャ・ページ I/O

このビューは、最新のサンプル間隔においてストアド・プロシージャの実行中に発生したページ I/O を示します。サンプル間隔においてページ I/O を生成した各ストアド・プロシージャについて、ストアド・プロシージャの名前と ID が、プロシージャが属するデータベースの名前と ID とともに表示されます。アクティブなストアド・プロシージャがないときにページ I/O が生成されると、これらの I/O のプロシージャ ID とデータベース ID の値はゼロになります。

このビューには、ストアド・プロシージャ・レベルごとに、最新のサンプル間隔においてストアド・プロシージャの実行中に生成されたページ I/O 回数、Adaptive Server のデータ・キャッシュで対応できたページ I/O 要求の割合、論理ページ読み込み回数、物理ページ読み込み回数、ページ書き込み回数も表示されます。

```
hs_create_view procedure_page_cache_io,
"Procedure Database Name", "Value for Sample", "Procedure
Database ID", "Value for Sample", "Procedure Name", "Value for
Sample", "Procedure ID", "Value for Sample", "Page I/O", "Value
for Sample", "Page Hit Percent", "Value for Sample", "Logical
Page Reads", "Value for Sample", "Physical Page Reads", "Value
for Sample", "Page Writes", "Value for Sample"
```

## プロセス・アクティビティ

このビューは、Adaptive Server のすべてのプロセスについて CPU 使用率、ページ I/O、現在のプロセスのステータスを示します。

最新のサンプル間隔について、各プロセスのログイン名、プロセス ID、カーネル・プロセス ID が、現在のプロセスのステータスとともに表示されます。

レコーディング・セッション開始以降の各プロセスの接続経過時間、ページ I/O 回数、CPU 利用時間も示されます。

```
hs_create_view process_activity,
"Login Name", "Value for Sample",
"Process ID", "Value for Sample",
"Kernel Process ID", "Value for Sample",
"Connect Time", "Value for Session",
"Page I/O", "Value for Session",
"CPU Time", "Value for Session",
"Current Process State", "Value for Sample"
```

## プロセスのデータベース・オブジェクト・ページ I/O

このビューは、各 Adaptive Server プロセスのデータベース・オブジェクト別ページ I/O を示します。最新のサンプル間隔においてページ I/O を生成した各プロセスについて、ログイン名、プロセス ID、カーネル・プロセス ID が表示されます。

これらの各プロセスとプロセスがアクセスした各データベース・オブジェクトについて、オブジェクト名、オブジェクト ID、オブジェクトのデータベース名と ID、オブジェクトと対応するページ I/O も表示されます。

また、最新のサンプル間隔について、ページ I/O 回数、Adaptive Server キャッシュで対応できたページ I/O 要求の割合、論理ページ読み込み回数、物理ページ読み込み回数、ページ書き込み回数も表示されます。

```
hs_create_view process_object_page_io,
"Login Name", "Value for Sample",
"Process ID", "Value for Sample",
"Kernel Process ID", "Value for Sample",
"Database Name", "Value for Sample",
"Database ID", "Value for Sample",
"Object Name", "Value for Sample",
"Object ID", "Value for Sample",
"Object Type", "Value for Sample",
"Page I/O", "Value for Sample",
"Page Hit Percent", "Value for Sample",
"Logical Page Reads", "Value for Sample",
"Physical Page Reads", "Value for Sample",
"Page Writes", "Value for Sample"
```

## プロセスのロック詳細情報

このビューは、最新のサンプル・インターバルが終了した時点で、Adaptive Server プロセスによって保持または要求されているデータベース・オブジェクトのロック・ステータスを示します。各ロックは、ロックされているオブジェクトの名前と ID、そのオブジェクトが属するデータベースの名前と ID、ロックを適用するページ番号（ページ・ロックの場合）とともに、ロックに関連付けられた Adaptive Server プロセスのログイン名、プロセス ID、カーネル・プロセス ID によって識別されます。各ロックの現在のステータスが表示され、デマンド・ロックであるかどうかが示されます。プロセスがロックを要求している場合は、このプロセスがロックの取得に待機した時間とロックを保持しているプロセスのプロセス ID が表示されます。プロセスがすでにロックを保持している場合は、そのロックを待機中のプロセス数が表示されます。

```
hs_create_view process_detail_locks,
"Login Name", "Value for Sample",
"Process ID", "Value for Sample",
"Kernel Process ID", "Value for Sample",
"Database Name", "Value for Sample",
"Database ID", "Value for Sample",
"Object Name", "Value for Sample",
"Object ID", "Value for Sample",
"Page Number", "Value for Sample",
"Lock Status", "Value for Sample",
"Demand Lock", "Value for Sample",
"Time Waited on Lock", "Value for Sample",
"Blocking Process ID", "Value for Sample",
"Locks Being Blocked Count", "Value for Sample"
```

## プロセスのページ I/O 詳細情報

このビューは、各 Adaptive Server プロセスのページ I/O を詳細に示します。最新のサンプル間隔が終了した時点での、各 Adaptive Server プロセスのログイン名、プロセス ID、カーネル・プロセス ID、現在のプロセスのステータス、現在のエンジンが表示されます。このビューは、サンプル間隔とセッションの開始以降の両方について、Adaptive Server データ・キャッシュで対応できたページ I/O 要求の割合を示します。

レコーディング・セッション開始以降から累積された論理ページ読み込み回数、物理ページ読み込み回数、ページ書き込み回数も表示されます。

```
hs_create_view process_detail_io,
"Login Name", "Value for Sample",
"Process ID", "Value for Sample",
"Kernel Process ID", "Value for Sample",
"Current Process State", "Value for Sample",
"Current Engine", "Value for Sample",
```

```

"Connect Time", "Value for Session",
"CPU Time", "Value for Session",
"Page Hit Percent", "Value for Sample",
"Page Hit Percent", "Value for Session",
"Logical Page Reads", "Value for Session",
"Physical Page Reads", "Value for Session",
"Page Writes", "Value for Session"

```

## プロセスのロック情報

このビューは、最新のサンプル・インターバル中にロック要求を生成した Adaptive Server の各プロセスのロック要求数を示します。

```

hs_create_view process_lock,
"Login Name", "Value for Sample",
"Process ID", "Value for Sample",
"Kernel Process ID", "Value for Sample",
"Lock Count", "Value for Sample"

```

## プロセスのページ I/O 情報

このビューは、最新のサンプルについて、Adaptive Server プロセス・レベル単位で要約されたページ I/O を示します。サンプル間隔においてページ I/O を生成した Adaptive Server の各プロセスについて、ログイン名、プロセス ID、カーネル・プロセス ID が表示されます。

このビューには、プロセス・レベルごとに、最新のサンプル間隔でのページ I/O 回数、Adaptive Server データ・キャッシュで対応できたページ I/O 要求の割合、論理ページ読み込み回数、物理ページ読み込み回数、ページ書き込み回数も表示されます。

```

hs_create_view process_page_io,
"Login Name", "Value for Sample",
"Process ID", "Value for Sample",
"Kernel Process ID", "Value for Sample",
"Page I/O", "Value for Sample",
"Page Hit Percent", "Value for Sample",
"Logical Page Reads", "Value for Sample",
"Physical Page Reads", "Value for Sample",
"Page Writes", "Value for Sample"

```

## プロセス・ステータスの概要

このビューは、最新のサンプル間隔が終了した時点での各プロセス・ステータスにあるプロセス数を表示します。

```
hs_create_view process_perf_sum,  
"Process State", "Value for Sample",  
"Process State Count", "Value for Sample"
```

## プロセスのストアド・プロシージャ・ページ I/O

このビューは、Adaptive Server プロセスによるストアド・プロシージャの実行に関連付けられたページ I/O を示します。サンプル間隔においてページ I/O を生成した各プロセスについて、ログイン名、プロセス ID、カーネル・プロセス ID が表示されます。ページ I/O を生成したプロセスとストアド・プロシージャごとに、ストアド・プロシージャが属するデータベースの名前と ID、プロシージャ自体の名前と ID も表示されます。

また、最新のサンプル間隔について、ページ I/O 回数、データ・キャッシュで対応できたページ I/O 要求の割合、論理ページ読み込み回数、物理ページ読み込み回数、ページ書き込み回数も示します。

```
hs_create_view process_procedure_page_io,  
"Login Name", "Value for Sample",  
"Process ID", "Value for Sample",  
"Kernel Process ID", "Value for Sample",  
"Procedure Database Name", "Value for Sample",  
"Procedure Database ID", "Value for Sample",  
"Procedure Name", "Value for Sample",  
"Procedure ID", "Value for Sample",  
"Page I/O", "Value for Sample",  
"Page Hit Percent", "Value for Sample",  
"Logical Page Reads", "Value for Sample",  
"Physical Page Reads", "Value for Sample",  
"Page Writes", "Value for Sample"
```

## サーバ・パフォーマンスの概要

このビューは、Adaptive Server の全体的なパフォーマンスを示します。最新のサンプル・インターバルにおける、1 秒間当たりのロック要求数、Adaptive Server がビジー状態の場合のサンプル・インターバルの割合、1 秒間当たりのトランザクション処理数、Adaptive Server がデッドロック回数が表示されます。

```
hs_create_view server_perf_sum,
"Lock Count", "Rate for Sample",
"CPU Busy Percent", "Value for Sample",
"Transactions", "Rate for Sample",
"Deadlock Count", "Value for Sample"
```

## ストアド・プロシージャ・アクティビティ

このビューは、プロシージャ文レベルのストアド・プロシージャ・アクティビティを示します。最新のサンプル間隔において実行されたストアド・プロシージャの各文は、プロシージャが属するデータベースの名前と ID、プロシージャの名前と ID、プロシージャ内の文の相対番号、プロシージャ文の先頭にあるテキスト行によって識別されます。

ビューには、最新のサンプル・インターバル間とレコーディング・セッション開始以降の両方について、各文の実行された回数が表示されます。サンプル・インターバルとレコーディング・セッションの両方における、文の実行に要した平均経過時間も表示されます。

```
hs_create_view procedure_activity,
"Procedure Database ID", "Value for Sample",
"Procedure Database Name", "Value for Sample",
"Procedure ID", "Value for Sample",
"Procedure Name", "Value for Sample",
"Procedure Line Number", "Value for Sample",
"Procedure Statement Number", "Value for Sample",
"Procedure Execution Count", "Value for Sample",
"Procedure Execution Count", "Value for Session",
"Procedure Elapsed Time", "Avg for Sample",
"Procedure Elapsed Time", "Avg for Session"
```

## トランザクション・アクティビティ

このビューは、サンプル・インターバルとレコーディング・セッションの両方における、Adaptive Server で発生したトランザクション・アクティビティの詳細を示します。

```
hs_create_view transaction_activity,  
"Transactions", "Value for Sample",  
"Rows Deleted", "Value for Sample",  
"Rows Inserted", "Value for Sample",  
"Rows Updated", "Value for Sample",  
"Rows Updated Directly", "Value for Sample",  
"Transactions", "Value for Session",  
"Rows Deleted", "Value for Session",  
"Rows Inserted", "Value for Session",  
"Rows Updated", "Value for Session",  
"Rows Updated Directly", "Value for Session",  
"Transactions", "Rate for Sample",  
"Rows Deleted", "Rate for Sample",  
"Rows Inserted", "Rate for Sample",  
"Rows Updated", "Rate for Sample",  
"Rows Updated Directly", "Rate for Sample",  
"Transactions", "Rate for Session",  
"Rows Deleted", "Rate for Session",  
"Rows Inserted", "Rate for Session",  
"Rows Updated", "Rate for Session",  
"Rows Updated Directly", "Rate for Session"
```

# 索引

## 記号

\$DSLISTEN 22  
\$SYBASE 12, 21, 33  
() (カッコ)  
  SQL 文内 xvi  
, (カンマ)  
  SQL 文内 xvi  
.bat ファイル 31  
::= (BNF 表記)  
  SQL 文内 xvi  
[] (角カッコ)  
  SQL 文内 xvi  
{ } (中カッコ)

## A

actual プレイバック 57, 154  
Adaptive Server 3, 33, 65

## B

Backus Naur Form (BNF) 表記 xvi  
bcp ユーティリティ 4, 86  
  例 94, 96  
BNF 表記、SQL 文内 xvi

## C

charsets ディレクトリ 34  
cut ユーティリティ 96

## D

-D パラメータ 10, 21, 65, 86  
dsep ユーティリティ 13  
dsedit ユーティリティ 13, 17  
DSLISTEN 環境変数 22

## E

entire プレイバック 57, 155

## H

Historical Server  
  Monitor Client Library 2  
  プレイバック・データ 2  
histserver コマンド 12, 20, 33  
histsvr コマンド 20, 31, 37  
hs.ctl ファイル 「制御ファイル」 参照  
hs.log ファイル 「ログ・ファイル」 参照  
hs\_create\_alarm コマンド 3, 48, 70  
hs\_create\_filter コマンド 3, 52, 70  
hs\_create\_playback\_session コマンド 5, 55, 158, 159, 167  
hs\_create\_playback\_view コマンド 5, 63, 158, 159  
hs\_create\_recording\_session コマンド 3, 65, 70, 92  
hs\_create\_view コマンド 3, 68, 70  
hs\_delete\_data コマンド 69  
hs\_initiate\_playback コマンド 5, 60, 61, 70  
hs\_initiate\_recording コマンド 3, 47, 70  
hs\_list コマンド 3, 10, 23, 61, 63, 71, 86  
hs\_playback\_sample コマンド 5, 59, 75  
hs\_recording コマンド 68  
hs\_shutdown コマンド 35, 38, 79  
hs\_status コマンド 24, 81  
hs\_terminate\_playback コマンド 5, 47, 62, 82  
hs\_terminate\_recording コマンド 47, 70, 82

## I

-i パラメータ 21  
interfaces ファイル 12, 13, 14, 21

## L

-l パラメータ 21, 34  
libtl.cfg ファイル 17  
locales ディレクトリ 34

# 索引

## M

Monitor Client Library 2  
  Historical Server 2  
Monitor Server 1, 33  
  Adaptive Server 2  
  接続 3  
Monitor Server 名 65  
Monitor Viewer 1, 2, 3  
monserver コマンド 13

## N

-n パラメータ 22  
no\_wait  
  「hs\_shutdown コマンド」参照  
no\_wait 「hs\_shutdown コマンド」参照

## O

ocscfg ユーティリティ 17  
Open Server 1

## P

-P パラメータ 22, 34, 38  
Procedure CPU Time 166  
Procedure Elapsed Time 166

## R

raw プレイバック 57, 153, 158  
regedit32.exe 25, 27

## S

-S パラメータ 22, 45  
SIGKILL シグナル 80  
sql.ini ファイル 12, 16  
Sybase Central 2, 3  
SYBASE 環境変数 12, 21, 33

## U

-U パラメータ 11, 22, 34, 38  
UNIX  
  Historical Server の起動 33  
  Historical Server の設定 12  
  第2サーバの設定 24  
  停止 34

## V

-v パラメータ 22

## W

Windows  
  Historical Server の設定 15  
  確認 37  
  起動パラメータの推定 37  
  第2サーバの設定 25  
  停止 38  
  レジストリ 25  
Windows レジストリ 27  
  起動パラメータ 37

## あ

アカウント  
  アラーム 49  
  起動 11, 23, 33  
  スーパユーザ 11, 22  
アクティビティ  
  現在のパーミッションの表示 81  
  なし 121  
アクティブなレコーディング・セッション 73  
アプリケーション・プログラミング・インターフェース 2  
アラーム  
  定義 48  
  定義のリスト 71  
  ログ・ファイルのエントリ 50  
  アラーム制御レコード、制御ファイル 90

## い

インスタンス 「複数インスタンス」 参照  
 インストール  
   ディレクトリ 12  
   手順 12, 16  
 インターバル、プレイバック・セッション 58, 156  
 引用符 43

## う

埋め込みスペース 43

## え

エラー  
   ステータス・コード 43  
   ファイル 92  
   メッセージ 43  
   レコーディング・セッション用オプション 67  
 エラー「ログ・ファイル」参照

## お

大文字と小文字の区別  
 SQL xvii

## か

開始時刻  
   プレイバック・セッション 55, 78, 154, 155  
   レコーディング・セッション 65  
 角カッコ []  
   SQL 文内 xvi  
 角カッコ。「角カッコ []」参照  
 カッコ ()  
   SQL 文内 xvi  
 空のロー 121  
 環境変数  
   DSLISTEN 22  
   SYBASE 12, 21, 33  
 カンマ ()  
   SQL 文内 xvi  
 管理 24

## き

キー・データ項目  
   定義 119  
 記号  
   SQL 文内 xvi  
 規則  
   Transact-SQL の構文 xvi  
   リファレンス・マニュアル xvi  
 起動  
   .bat ファイル 31  
   UNIX の場合 33  
   Windows の場合 37  
   アカウント 11, 23, 33  
   スクリプト・ファイル 12  
   パラメータ 20, 22  
   プレイバック・セッション 70  
 起動の確認 37  
 起動パラメータ  
   Windows レジストリから 37  
 起動パラメータの推定 37  
 規約  
   「構文」参照  
 競合、制御ファイル 24  
 共有メモリ 2

## &lt;

クライアント接続「接続」参照  
 クライアント・プレイバック 60, 75  
 グラフィカル・ユーザ・インターフェース 2

## け

計算、プレイバック・ビュー 159  
 結果データ項目、定義 119  
 結果、プレイバック 62

## ニ

構文規則、Transact-SQL xvi  
 コマンド  
   histserver 12, 20, 33  
   histsrvr 20, 31, 37  
   hs\_create\_alarm 3, 48, 70  
   hs\_create\_filter 3, 52, 70  
   hs\_create\_playback\_session 5, 55, 158, 159, 167  
   hs\_create\_playback\_view 5, 63, 158, 159  
   hs\_create\_recording\_session 3, 65, 70, 92  
   hs\_create\_view 3, 68, 70  
   hs\_delete\_data 69  
   hs\_initiate\_playback 5, 60, 61, 70  
   hs\_initiate\_recording 3, 47, 68, 70  
   hs\_list 3, 10, 23, 61, 63, 86  
   hs\_playback\_sample 5, 59, 75  
   hs\_shutdown 35, 38, 79  
   hs\_status 24, 81  
   hs\_terminate\_playback 5, 47, 62, 82  
   hs\_terminate\_recording 47, 70, 82  
 monserver 13  
 概要 41  
 構文 42  
 「ユーティリティ」参照

## さ

サーバ設定ユーティリティ 25  
 サーバ・レベル・ビュー 122  
 サービス・リスト 27  
 削除、データ・ファイル 61, 69  
 作成  
   アラーム 48  
   テーブル 61, 67, 93  
   フィルタ 52  
   プレイバック・セッション 47  
   プレイバック・ビュー 63, 153  
   レコーディング・セッション 3, 47  
   レコーディング・セッション・ビュー 68, 119  
 サンプル・インターバル 3, 23, 65  
 ユーザ定義、プレイバック 155

## し

シャットダウン 79  
 パーミッション 11  
 終了 79  
   UNIX の場合 34  
   Windows の場合 38  
   プレイバック・セッション 82  
 終了時刻  
   プレイバック・セッション 56, 78  
   レコーディング・セッション 66  
 出力、プレイバック 62  
 所有者、データ・ファイル 86

## す

推定、プレイバック・ビュー 59, 159  
 スーパユーザ 11, 22, 34, 38, 45, 81  
 スクリプト・ファイル 10  
   Adaptive Server でのテーブルの作成 92  
   アラーム 49  
   起動用 12  
   テーブル作成用 61, 67  
 ステータス  
   Historical Server 24, 81  
   セッション 73  
   レコーディング・セッション 73  
 ステータス・コード 43

## せ

制御ファイル 4, 10, 21, 23  
 起動アカウント 10  
 競合 24  
   セッション制御レコード 88  
   データ項目制御レコード 90  
   パーミッション 10  
   ビュー制御レコード 89  
   フィルタ制御レコード 91  
   フォーマット 86  
   ヘッダ・レコード 87  
   編集 10, 87  
   セッション ID 3  
   セッション制御レコード 88  
   セッションの終了 47, 82  
   セッション、相互に排他的な 62

## 接続

Historical Server 45  
 interfaces ファイル 12  
 Monitor Server 3, 45, 65  
 sql.ini ファイル 12  
 現在のパーミッションの表示 81  
 最大数 22, 81  
 相互に排他的なセッション 46  
 定義 47  
 ディレクトリ・サービス 12  
 パーミッション 45  
 接続の最大数 22, 81  
 設定

Historical Server 起動パラメータ 20, 22  
 Historical Server を UNIX 上で設定 12  
 Windows での Historical Server 15  
 サービス・リスト 27  
 第 2 Historical Server を UNIX 上で追加 24  
 第 2 Historical Server を Windows 上で追加 25

## そ

相互に排他的なセッション 46, 62

## た

タイムスタンプ 23, 165  
 タイム・ゾーン 56, 65

## ち

中カッコ {}, SQL 文内 xvi

## つ

追加  
 .bat ファイル 31  
 Windows レジストリ・レジストリ 25  
 サービス・リスト 27  
 第 2 Historical Server を UNIX 上で追加 24  
 第 2 Historical Server を Windows 上で追加 25

## て

定義「作成」参照  
 ディレクトリ  
 インストール 12  
 データ・ファイル 65, 86  
 ロケールと文字セット 34  
 「ホーム・ディレクトリ」参照  
 ディレクトリ・サービス 12  
 データ項目  
 Procedure CPU Time 166  
 Procedure Elapsed Time 166  
 タイムスタンプ 165  
 定義 103  
 プレイバック・ビューの要件 160  
 プロセス ID 166  
 有効な統計タイプ 145  
 レコーディング・セッションの要件 122  
 データ項目制御レコード 90  
 データ・ファイル 10, 85  
 削除 61, 69  
 所有者 86  
 内容の説明 4, 42, 85  
 パーミッション 10, 11, 24, 66, 86  
 フォーマット 91  
 ロケーション 65, 86  
 テーブル・カラムの名前、スクリプト・ファイル 93  
 テーブル、作成 67, 93

## と

統計タイプ 6  
 各データ項目 145  
 プレイバック・ビュー 160

## に

入力セッション「レコーディング・セッション」を参照  
 入力ビュー 158

## は

バージョン  
  Historical Server 22, 81

バーミッシュン  
  インストール・ディレクトリ 12  
  概要 45  
  シャットダウン 11  
  スーパーユーザ 45  
  制御ファイル 10  
  データ・ファイル 10, 11, 24, 86  
  ブレイバッック・ファイル 60  
  レコーディング・セッション・ファイル 66

排他的なセッション 62

パスワード  
  起動コマンド 22  
  プロンプト 22

パラメータ  
  -D 10, 21, 65, 86  
  -i 21  
  -l 21, 34  
  -n 22  
  -P 22, 34, 38  
  -S 22, 45  
  -U 11, 22, 34, 38  
  -v 22  
  起動 20, 22

バルク・コピー・ユーティリティ 「bcp ユーティリティ」参照

## ひ

非アクティブなレコーディング・セッション 73

ビュー  
  概要 5, 119  
  サーバ・レベル 122  
  定義のリスト 71  
  ブレイバッック 5  
  ブレイバッック・ビューの要件 160  
  「レコーディング・セッション・ビュー」参照

ビュー制御レコード 89

ビューの例 169

## ふ

ファイル  
  .bat 31  
  interfaces 12, 13, 14, 21  
  libctl.cfg 17  
  sql.ini 12, 16  
  データ 85  
  ブレイバッック・ターゲットとしてのファイル 60  
  「制御ファイル」参照

フィルタ 52  
  定義のリスト 71

フィルタ値の指定 52, 53

フィルタ制御レコード 91

複数インスタンス 23  
  UNIX の場合 24  
  Windows の場合 25

ブレイバッック 2  
  インターバル 58, 156  
  データのギャップ 63

ブレイバッック・セッション  
  開始時刻 78  
  概要 4  
  起動 70  
  クライアント 75  
  結果 62  
  作成 47  
  サンプル・インターバル 155  
  終了 82  
  終了時刻 78  
  相互排他的 46  
  データのギャップ 78  
  ファイルのバーミッシュン 60  
  要約のリスト 71

ブレイバッック・セッションの開始 70

ブレイバッック・セッションのターゲット 60, 62

ブレイバッック・データのギャップ 63, 78

ブレイバッック・ビュー 5  
  計算 159  
  作成 63, 153  
  使用条件 160  
  推定 159  
  名前 63  
  有効な統計タイプ 160  
  要約レベル 153  
  プロセス ID 166

## へ

平均、統計タイプ  
定義 7  
ヘッダ・レコード 87  
編集、制御ファイル 10, 87

## ほ

ホーム・ディレクトリ 4, 10, 21  
起動アカウント 11

## ιΦ

ユーザ定義サンプル・インターバル 155  
ユーザ名 22  
ユーティリティ  
  bcp 86  
  cut 96  
  dscep 13  
  dsedit 13, 17  
  ocscfg 17  
  サーバ設定 25

## よ

要約レベル 71, 153

## り

リストイング  
  アラーム 71  
  ビュー 71  
  フィルタ 71  
  要約レベル 71  
  レコーディング・セッション定義 71

## れ

レコーディング・セッション  
  エラー 67  
  開始時刻 65  
  概要 3  
  過去 71  
  起動 70  
  作成 3, 47  
  終了 70, 82  
  終了時刻 66  
  ステータス 73  
  相互排他的 46  
  定義のリスト 71  
  データの調査 4  
  同時 22  
レコーディング・セッション・ビュー  
  作成 68  
  使用条件 122  
  名前 68  
  例 169  
  レコード機能 3

## ろ

ロー、空 121  
ログ・ファイル 10, 21, 34  
  アラーム 49, 50  
  名前 81  
  ロケーション 34

