

# CaoSQL Tools

## ユーザーズ ガイド

Version 1.8.0

December 14, 2013

【備考】

**【改版履歴】**

日付	版数	内容
2006-02-24	1.0.0	初版.
2011-02-22	1.1.0	CaoSQL プロバイダに, 特殊コントローラ名情報を追加
2013-12-14	1.8.0	Execute コマンド追加, ID プロパティ追加

## 目次

1. はじめに .....	5
2. CaoSQLTester .....	6
2.1. 概要 .....	6
2.2. 操作方法 .....	6
2.2.1. CaoSQLEngine .....	7
2.2.2. CaoSQLController .....	8
2.2.3. CaoSQLItem .....	11
2.2.4. CaoSQLHistory .....	14
3. CaoSQLCmd .....	15
3.1. 概要 .....	15
3.2. 機能 .....	15
4. CaoSQLLauncher .....	18
4.1. 概要 .....	18
4.2. 操作方法 .....	18
4.3. コマンドライン引数 .....	18
5. CaoSQL プロバイダ .....	19
5.1. はじめに .....	19
5.2. 概要 .....	19
5.3. CaoProvCAOSQL.dll の使い方 .....	20
5.3.1. はじめに .....	20
5.3.2. CaoWorkspace::AddController メソッド .....	21
5.3.3. CaoController::AddVariable メソッド .....	22
5.3.4. CaoController::Execute メソッド .....	22
5.3.5. CaoController::get_Help プロパティ .....	23
5.3.6. CaoController::get_ID プロパティ .....	23
5.3.7. CaoController::get_Tag プロパティ .....	23
5.3.8. CaoController::put_Tag プロパティ .....	23
5.3.9. CaoController::get_VariableNames プロパティ .....	23
5.3.10. CaoVariable::get_Attribute プロパティ .....	24
5.3.11. CaoVariable::get_DateTime プロパティ .....	24
5.3.12. CaoVariable::get_Help プロパティ .....	24
5.3.13. CaoVariable::get_ID プロパティ .....	24
5.3.14. CaoVariable::put_ID プロパティ .....	24

---

5.3.15. CaoVariable::get_Microsecond プロパティ.....	24
5.3.16. CaoVariable::get_Tag プロパティ.....	24
5.3.17. CaoVariable::put_Tag プロパティ.....	24
5.3.18. CaoVariable::get_Value プロパティ.....	24
5.3.19. CaoVariable::put_Value プロパティ.....	24
<b>6. リンクビューア .....</b>	<b>25</b>
6.1. 概要.....	25
6.2. アドインの追加.....	26
6.3. リンクビューアの詳細 .....	29
6.3.1. リンク先の数表示.....	29
6.3.2. リンク先の不明によるエラー表示 .....	29
<b>7. タイミングチャート .....</b>	<b>31</b>
7.1. 概要.....	31
7.2. アドインの追加.....	32
7.3. タイミングチャートの使用方法.....	32
7.4. タイミングチャートの詳細.....	35
<b>8. jCaoSQL.....</b>	<b>37</b>
8.1. 概要.....	37
8.2. クライアント開発環境セットアップ .....	37
8.3. サンプル解説.....	37
8.3.1. インスタンスの生成と破棄.....	37
8.3.2. VARIANT 型の対応.....	38
8.3.3. イベントシンクの作成.....	38
8.3.4. カスタムクラスの作成 .....	38
8.3.5. アイテムの動的追加, 削除 .....	39

## 1. はじめに

本書は CaoSQL 専用ツールのユーザーズガイドです。<sup>1</sup>

CaoSQLTester ツールは、CaoSQL のすべてのインタフェースが実装されたクライアントアプリケーションで、CaoSQL のメソッドの実行やプロパティの取得・設定のテストが行えます。

CaoSQLLauncher ツールは、指定したマシンの CaoSQL.exe を起動するためのツールです。

CaoSQLCmd ツールは、CaoSQL の公開インタフェース経由で様々なコマンドを送ることができるコンソール・アプリケーションです。例えば、あるコントローラのサンプリングを一時停止するなど、コントローラ単位の制御も可能になります。また、Windows に付属のタスクスケジューラと組み合わせることで、ある時間だけサンプリングを行う設定などもできます。

リンクビューアドインツールは、CaoSQLConfig で設定された CaoSQLItem の LinkItem プロパティに設定されているリンク状態を、視覚的に図表示する Microsoft Excel のアドインツールです。これを用いることによって、ユーザはどのアイテムがどのアイテムにリンクしているか、などの全体の設定を容易に把握することができます。

タイミングチャートアドインツールは、CaoSQLConfig で設定された CaoSQLItem の値の変化をタイミングチャートに表示して、信号の ON/OFF のタイミングを容易に比較することができる Microsoft Excel のアドインツールです。これを用いることによって、ユーザはあるアイテムがどのタイミングで ON になり、どのタイミングで OFF になるかなどが容易に把握することができます。

jCaoSQL は java から CaoSQL のコントロールをするためのフレームワークで、CaoSQL のインタフェースを介して、ロボットコントローラなどのリソースにアクセスすることができます。

---

<sup>1</sup> CaoSQLConfig については『[CaoSQL ユーザーズガイド](#)』の「4 CaoSQLConfig」の章を参照して下さい。

## 2. CaoSQLTester

### 2.1. 概要

ここでは、CaoSQL をテストするためのツールである CaoSQLTester について解説します。

CaoSQLTester は、CaoSQLEngine, CaoSQLController, CaoSQLItem, CaoSQLHistory の持つメソッドとプロパティを実装しており、これらのテストを行うことができます。

### 2.2. 操作方法

画面右のスペースはツリービューのノードをダブルクリックすることによって、そのノードに対応したオブジェクト情報のダイアログを表示します。エンジンのノードをダブルクリックするとエンジンのダイアログが表示されます。コントローラのノードをダブルクリックするとそのコントローラのダイアログが表示されます。アイテムのノードをダブルクリックするとそのアイテムのダイアログが表示されます。

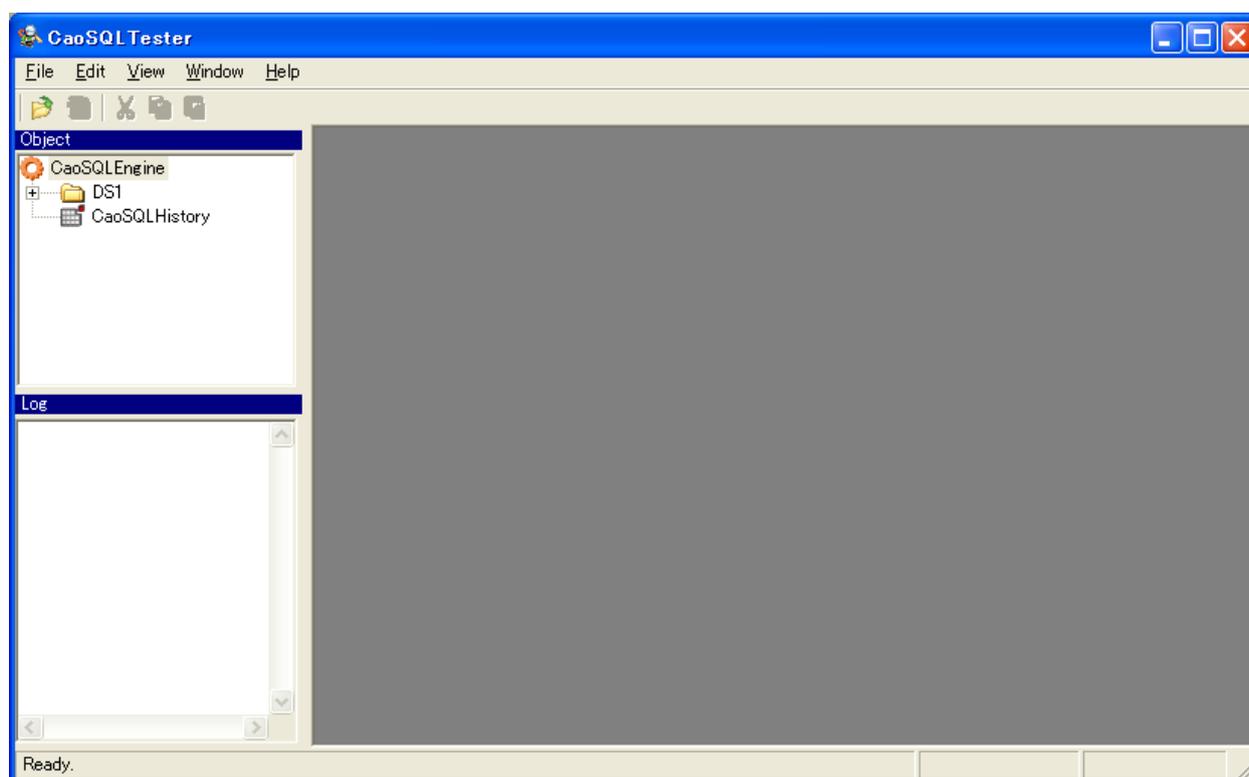


図 2-1 CaoSQLTester 起動画面

### 2.2.1. CaoSQLEngine

CaoSQL エンジンの機能を実行することができます。

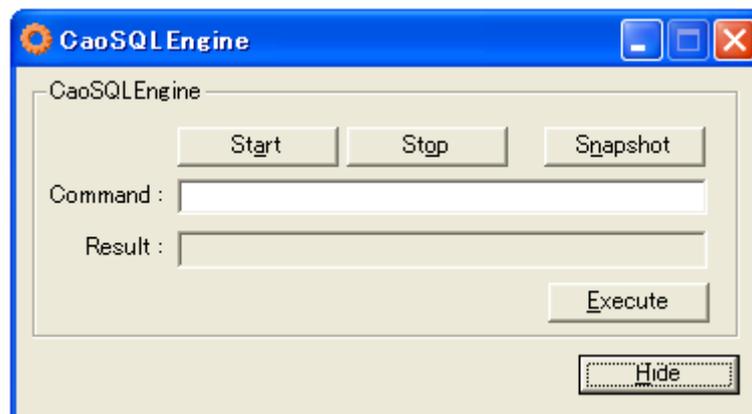


図 2-2 CaoSQLEngine ウィンドウ

#### [CaoSQL エンジンサンプリング開始] – Start

全コントローラのサンプリングを開始します。

#### [CaoSQL エンジンサンプリング停止] – Stop

全コントローラのサンプリングを停止します。

#### [CaoSQL エンジンスナップショット] – Snapshot

全コントローラ的全アイテムを 1 回だけサンプリングします。

#### [CaoSQL エンジンコマンド実行] – Execute<sup>2</sup>

CaoSQL エンジンの拡張コマンドを実行します。

<sup>2</sup> 現在、この機能は実装されていません。

## 2.2.2. CaoSQLController

CaoSQL コントローラの機能を実行することができます。

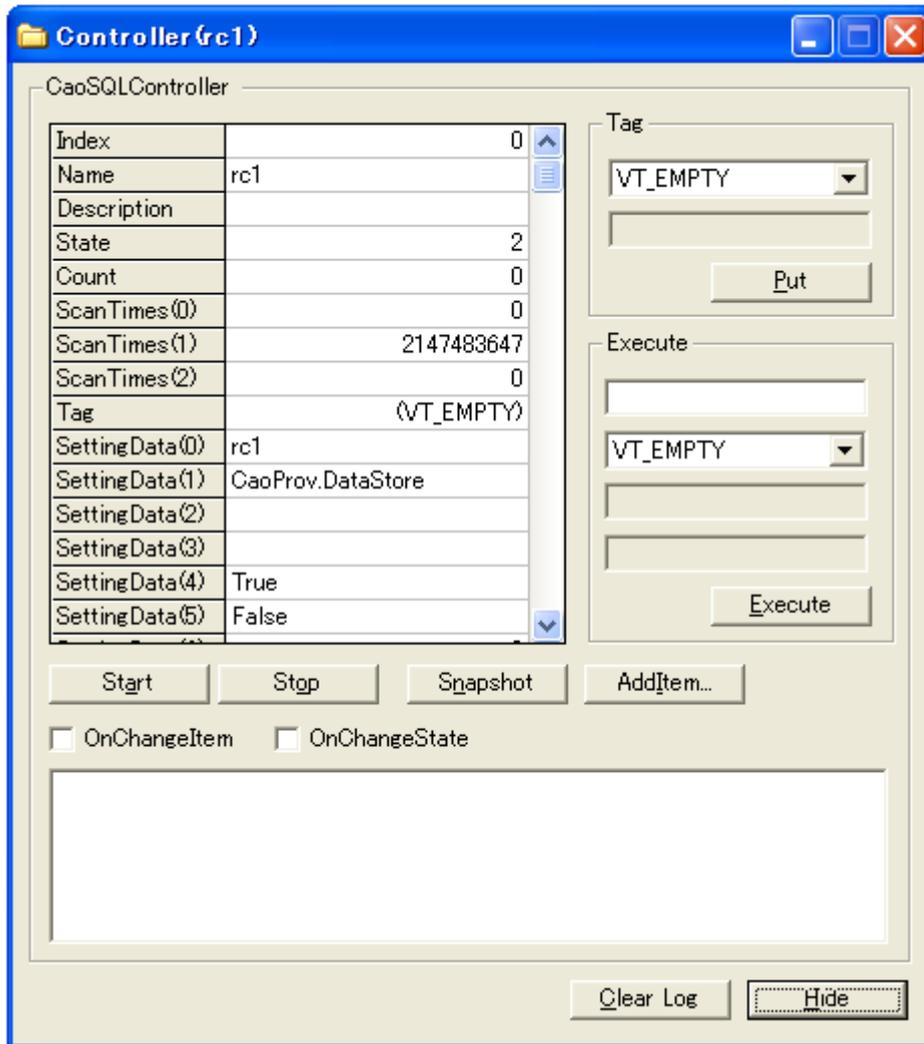


図 2-3 CaoSQLController ウィンドウ

### [CaoSQL コントローラサンプリング開始] – Start

コントローラ配下の全アイテムのサンプリングを開始します。

### [CaoSQL コントローラサンプリング停止] – Stop

コントローラ配下の全アイテムのサンプリングを停止します。

### [CaoSQL コントローラサンプリング] – Snapshot

コントローラ配下の全アイテムを 1 回だけサンプリングします。

**[CaoSQL コントローラタグ書き込み] – Put**

コントローラタグにデータを書き込みます。

データは RAC のデータ記述方式で指定します。<sup>3</sup>

以下に記入例を示します。

例 1)	Type:I2	Parameter: 100	値:100
例 2)	Type:BSTR	Parameter: Sample	値:“Sample”
例 3)	Type:VARIANT	Parameter: (8,Sample)	値:“Sample”
例 4)	Type:ARRAY I2	Parameter: 100,200,300	値:100, 200, 300
例 5)	Type:ARRAY VARIANT	Parameter: (8,Sample),(2,100)	値:“Sample”, 100

**[CaoSQL コントローラ コマンド実行] – Execute**

指定されたコマンドを実行します。

パラメータは RAC のデータ記述方式で指定します。<sup>3</sup>

**[CaoSQL コントローラ アイテム変化イベント] – OnChangeItem**

このチェックボックスをチェックすると、登録されたアイテムの値が変化した時に OnChangeItem イベントが発行され、下部のログウインドウに OnChangeItem イベントが発生したアイテム名とアイテム値を表示します。

**[CaoSQL コントローラ 状態変化イベント] – OnChangeState**

このチェックボックスをチェックすると、登録されたコントローラのスレータスの値が変化した時に OnChangeState イベントが発行され、下部のログウインドウにコントローラスレータス値を表示します。

**[CaoSQL コントローラアイテム追加] – AddItem**

CaoSQL コントローラにアイテムを追加します。

設定項目は、CaoSQLConfig と同じですので、設定内容については、『[CaoSQL ユーザーズガイド](#)』の「4 CaoSQLConfig」の章を参照してください。

<sup>3</sup>詳細については『RAC ユーザーズガイド』の「3.2 データの記述方式」を参照してください。

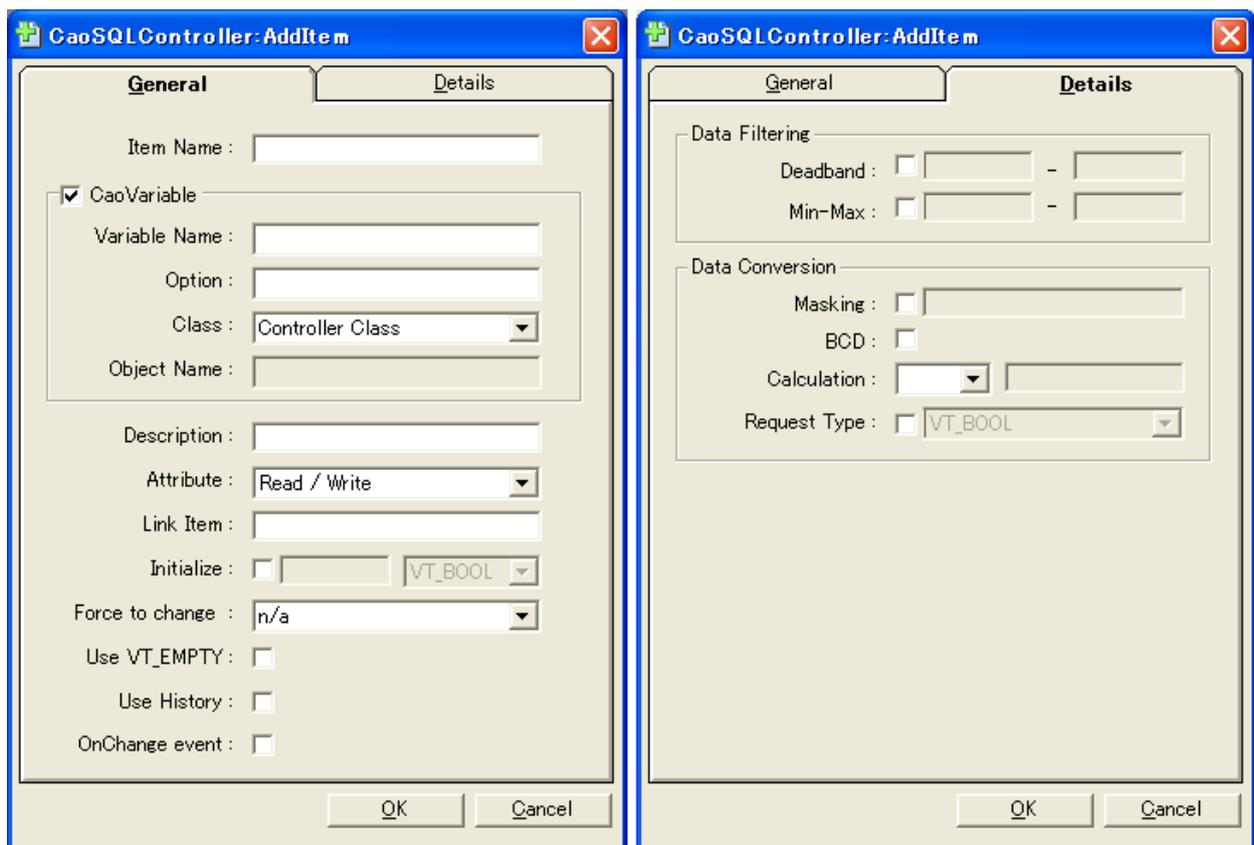


図 2-4 アイテムの追加

### 2.2.3. CaoSQLItem

CaoSQL アイテムの機能を実行することができます。

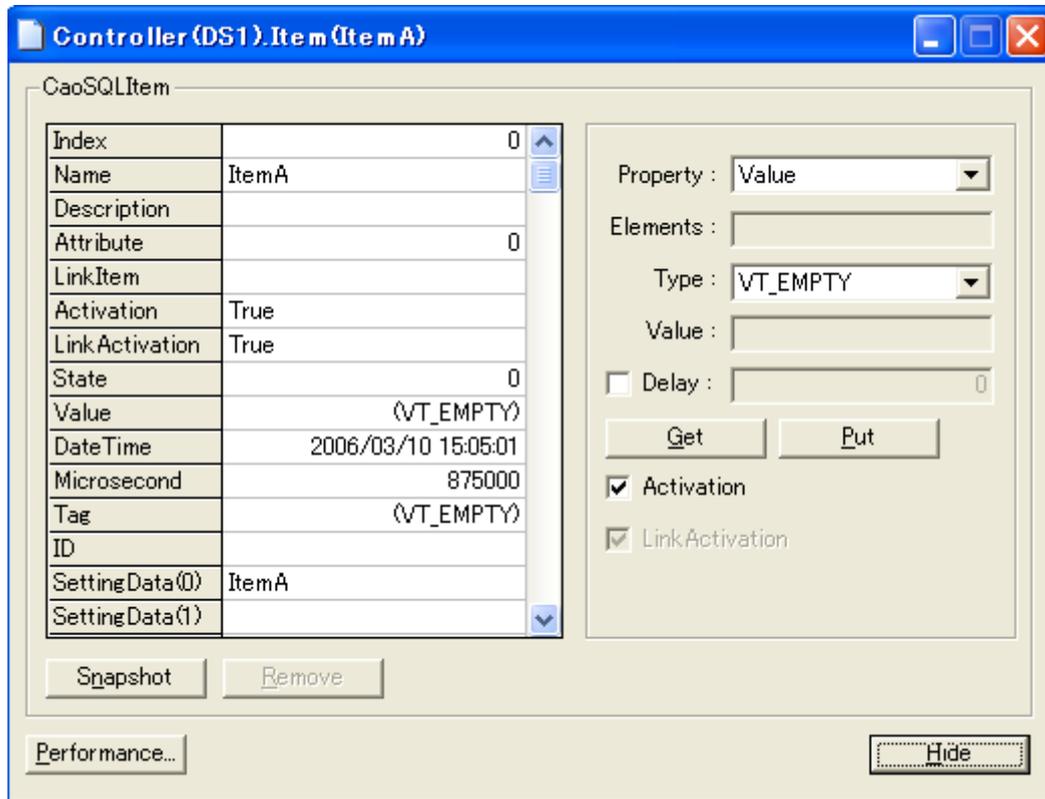


図 2-5 CaoSQLItem ウィンドウ

#### [CaoSQL アイテム スナップショット] – Snapshot

アイテムを 1 回だけサンプリングします。

#### [CaoSQL アイテム 削除] – Remove

アイテムが動的追加アイテムの場合、アイテムを削除します。

アイテムが動的追加アイテムではない場合、この処理は行えません。

#### [CaoSQL アイテム 値/タグ] – Value / Tag

アイテムの Put する先を、値かタグから選択します。

#### [CaoSQL アイテム 書き込み] – Put

値または、アイテムタグにデータを書き込みます。

データは RAC のデータ記述方式で指定します。<sup>3</sup>

**[CaoSQL アイテム 活性化] – Activation**

チェックボックスをオンにすると、アイテムを活性化します。

**[CaoSQL アイテム 値伝播機能] – LinkActivation**

チェックボックスをオンにすると、アイテムリンク先への値伝播機能を有効にします。

**[CaoSQL アイテム パフォーマンスチェック] – Performance**

アイテムの読み書き性能のチェックを行います。

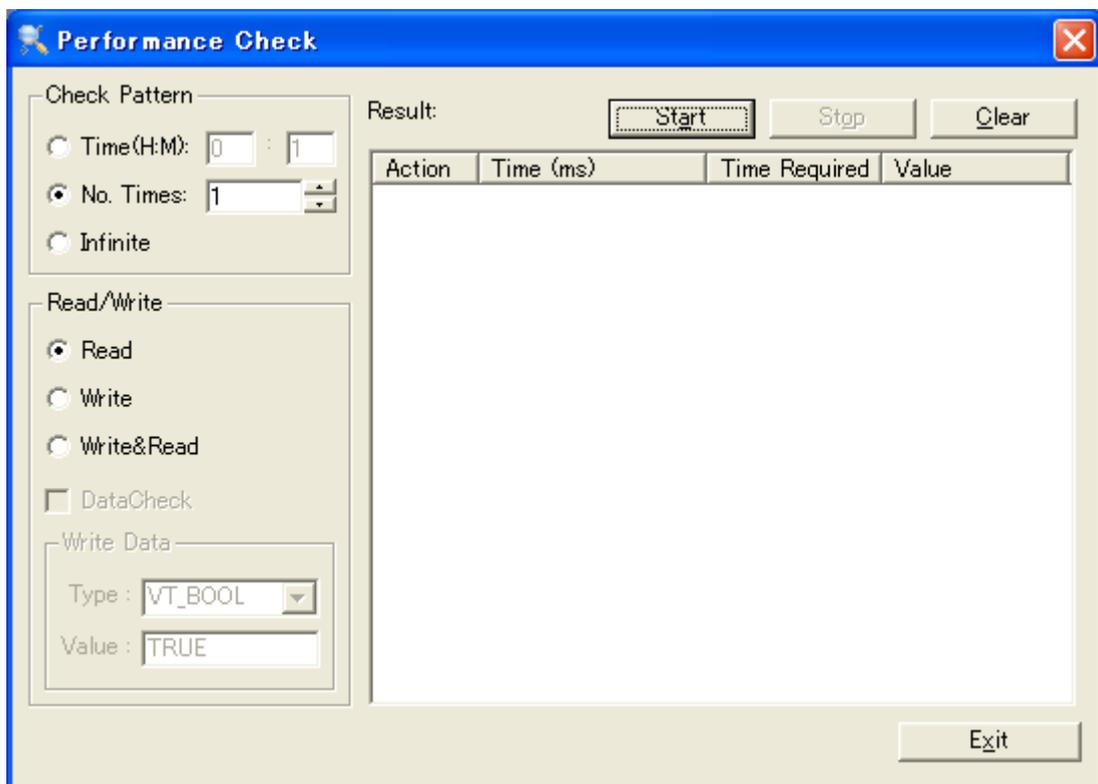


図 2-6 Performance Check 画面

**[チェックパターン] – Time / No. Times / Infinite**

パフォーマンスチェックを行う時間もしくは、回数を設定します。設定項目を以下に示します。

- Time** : 時間で設定します。単位は、時:分です。
- No. Times** : 回数で設定します。
- Infinite** : Stop ボタンを押下するまで、無限にチェックを行います。

**[チェック項目] – Read / Write / Write & Read**

パフォーマンスチェックを行うチェック項目の設定を行います。設定項目を以下に示します。

- Read** : 読み込みのみのパフォーマンスチェックを行います。
- Write** : 書き込みのみのパフォーマンスチェックを行います。
- Write & Read** : 書き込みと読み込みのパフォーマンスチェックを行います。

#### [データチェック] – DataCheck

書き込んだデータと読み込んだデータの値が等しいか同化のチェックを行います。チェック項目に“Write & Read”を選択した場合にのみ設定することができます。

書き込んだデータと読み込んだデータの値が異なった場合は、ダイアログによりエラーが通知されます。

#### [書き込みデータ] – WriteData

書き込むデータの型と値を設定することができます。チェック項目に“Write”または、“Write & Read”を選択した場合にのみ設定することができます。

#### [パフォーマンスチェック 開始] – Start

パフォーマンスチェックを開始します。

#### [パフォーマンスチェック 停止] – Stop

パフォーマンスチェックを停止します。

#### [ログのクリア] – Clear

テスト結果をクリアします。

## 2.2.4. CaoSQLHistory

CaoSQL ヒストリの機能を実行することができます。

ヒストリが無効の場合、SQL Command テキストボックスと Execute ボタン、Result グリッドが使用不可になります。

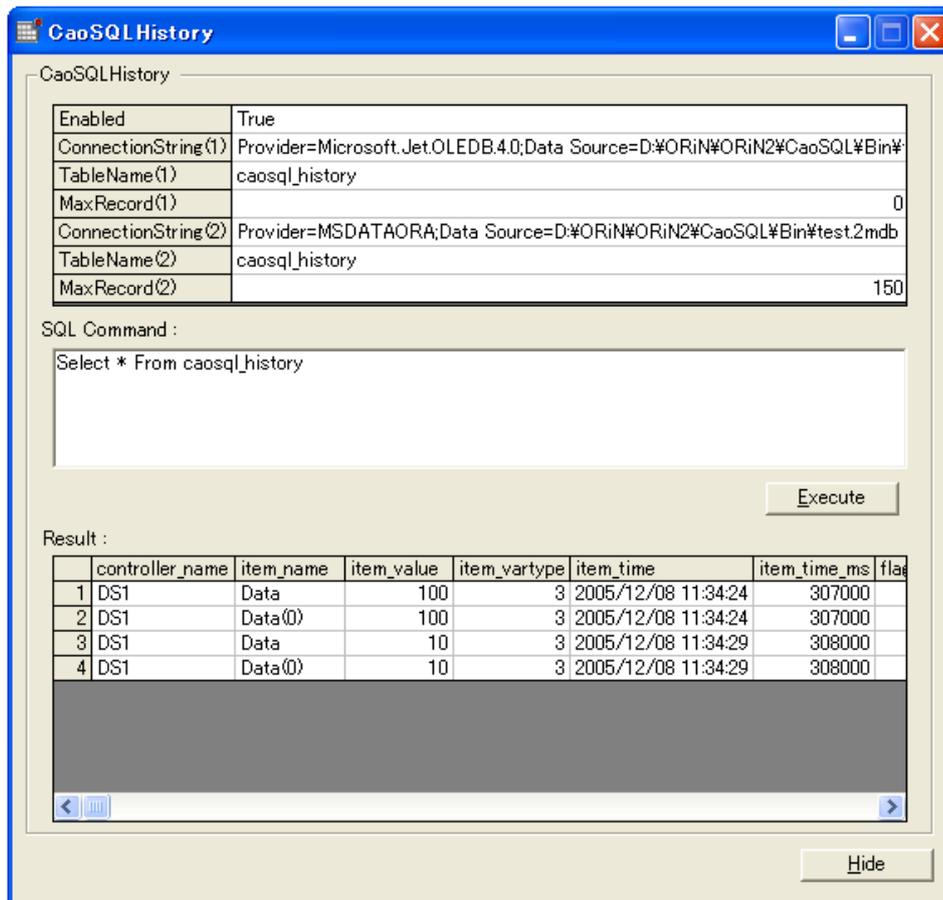


図 2-7 CaoSQLHistory ウィンドウ

### [CaoSQL ヒストリ SQL 文入力] – SQL Command

実行したい SQL コマンドを入力します。

### [CaoSQL ヒストリ SQL 文実行] – Execute

SQL Command テキストボックス内に記述された SQL 文を実行します。

### [CaoSQL ヒストリ SQL 文実行結果] – Result

Execute ボタンで実行した SQL 文の実行結果をグリッド表示します。

## 3. CaoSQLCmd

### 3.1. 概要

ここではCaoSQLをコマンドラインからコントロールするツールとしてCaoSQLCmd.exeを解説します。このツールは主にタイムベースでCaoSQLの動作を管理する目的で作成されました。CaoSQLCmdはコンソールアプリケーションであり、このツールをWindows NT/2000/XP等にOS標準で搭載されている「タスクマネージャ」からバッチ処理で使用することによってCaoSQLの時間管理が容易に行えます。

また、/Value /put コマンド(後述)とUPSのシャットダウン処理と同期させれば、任意のタイミングでCaoSQLアプリケーションに簡単にシャットダウンを通知でき、アプリケーションは安全に処理を終了させることができます。

### 3.2. 機能

CaoSQLCmdは以下のコマンドが使用できます。これらのコマンドは任意に組み合わせることができ、以下の文法に従ってCaoSQLに対して様々な処理を実行することができます。

- (1) /Start コマンドの場合:

```
CaoSQLCmd /Start [<コマンド対象コントローラ>]
```

- (2) /Stop コマンドの場合:

```
CaoSQLCmd /Stop [<コマンド対象コントローラ>]
```

- (3) /Snapshot コマンドの場合:

```
CaoSQLCmd /Snapshot [<コマンド対象コントローラ>[<コマンド対象アイテム>]]
```

- (4) /State コマンドの場合:

```
CaoSQLCmd /State <対象コントローラ>[<対象アイテム>]
```

- (5) /Value コマンドの場合:

```
CaoSQLCmd /Value <対象コントローラ> <対象アイテム>[ /Put <データ型>, <値>]
```

ここで、コマンド名やコントローラ名は大文字・小文字の区別はありません。[]は省略可能であることを意味しています。また、コマンド対象コントローラ/アイテムは省略可能で、省略した場合は全てのコントローラ/アイテムに対してコマンドが実行されます。すべてのコマンドは、すでに実行済みのCaoSQLに対して実行するもので、CaoSQLが実行されていない状態では必ずエラーを返します。/Value コマンドで/Putを使用する場合、値にスペースを含む場合は値をダブルクォーテーションで括り、データの書式は『[ORiN2 プログラミングガイド](#)』の「2.2.5.データの記述方式」を参照して下さい。

表 3-1 CaoSQLCmd のコマンド実行例一覧

コマンド	例	意味
/Start	CaoSQLCmd.exe /Start	全コントローラのスレッドを開始します。
	CaoSQLCmd.exe /Start RC1	対象コントローラのスレッドを開始します。
/Stop	CaoSQLCmd.exe /Stop	全コントローラのスレッドを停止します。
	CaoSQLCmd.exe /Stop RC1	対象コントローラのスレッドを停止します。
/Snapshot	CaoSQLCmd.exe /Snapshot	全コントローラのスナップショットを実行します。
	CaoSQLCmd.exe /Snapshot RC1	対象コントローラのスナップショットを実行します。
	CaoSQLCmd.exe /Snapshot RC1 ItemA	対象コントローラの任意のアイテムのスナップショットを実行します。
/State	CaoSQLCmd.exe /State RC1	対象コントローラの状態を取得します。
	CaoSQLCmd.exe /State RC1 ItemA	対象コントローラの任意のアイテムの状態を取得します。
/Value	CaoSQLCmd.exe /Value RC1 ItemA	対象コントローラの任意のアイテムの値を取得します。
/Value /Put	CaoSQLCmd.exe /Value RC1 ItemA /Put 8,Test	対象コントローラの任意のアイテムの値を設定します。

**/Start**

スタートコマンドは CaoSQLEngine/CaoSQLController インタフェースの Start メソッドを実行します。このコマンドは引数として、Start したいコントローラ名を指定することができます。

**/Stop**

ストップコマンドは CaoSQLEngine/CaoSQLController インタフェースの Stop メソッドを実行します。Start コマンド同様、対象コントローラ名を指定することもできます。

**/Snapshot**

スナップショットコマンドは CaoSQLEngine/CaoSQLController/CaoSQLItem インタフェースの Snapshot メソッドを実行します。対象コントローラ名および、対象アイテム名を指定することができます。

**/State**

ステータスコマンドは CaoSQLController/CaoSQLItem インタフェースの State プロパティを取得します。

対象コントローラ名, 対象アイテム名を指定することもできます.

### **/Value**

Value コマンドは CaoSQLItem インタフェースの `get_Value` メソッドを実行し, `/Put` オプションを付加すると `put_Value` メソッドを実行します. `/Put` オプションで設定するデータの書式は『[ORiN2 プログラミングガイド](#)』の「2.2.5.データの記述方式」を参照してください. Value コマンドの文法は上記のコマンドと少し違うので注意して下さい.

このコマンドの使用例として, UPS のシャットダウン処理と同期させて(要は「シャットダウン開始時に実行するファイル」として登録する), 適当なアイテムの値を変更すれば, アプリケーションは `OnChangeItem` イベントなどを用いてシャットダウンを容易に検知することができます.

## 4. CaoSQLLauncher

### 4.1. 概要

CaoSQLLauncher ツールは指定したマシンの CaoSQL.exe を起動するためのツールです。また、スタートアップに CaoSQLLauncher のショートカットを登録しておくことで特定のユーザだけに CaoSQL を起動させることができます。また、タスクトレイに常駐することもできます。

### 4.2. 操作方法



図 4-1 CaoSQLLauncher メイン画面

#### [ホスト名] – Host

起動したい CaoSQL.exe のあるマシン名, または IP アドレスを指定します。

#### [CaoSQL 起動] – Start

指定したホストの CaoSQL.exe を起動します。

#### [CaoSQL 終了] – Stop

起動した CaoSQL.exe を終了します。

### 4.3. コマンドライン引数

CapSQLLauncher は以下のコマンドライン引数を指定することができます。

```
GaoSQLLauncher.exe [ HIDE ] [ START [ <ホスト名> ] ]
```

- HIDE** : タスクトレイに格納した状態で起動します。
- START** : CaoSQL を起動した状態で起動します。
- <ホスト名>** : ホスト名を設定します。

例1) タスクトレイに格納, デフォルト(localhost)の CaoSQL を起動した状態で起動する。

```
CaoSQLLauncher.exe HIDE START
```

例2) ホスト名:TestServer の CaoSQL を起動した状態で起動する。

```
CaoSQLLauncher.exe START TestServer
```

## 5. CaoSQL プロバイダ

### 5.1. はじめに

ここでは CAO プロバイダの一つである CaoSQL プロバイダの説明をします。CaoSQL プロバイダは、CAO のクライアントツールである CaoSQL をより有効に活用する目的で開発されました。CaoSQL プロバイダは CaoSQL が随時収集するデータを CAO のインターフェースに従って取得することが可能です。この CaoSQL プロバイダが持つ機能により、分散環境下での CaoSQL の構成に幅を持たせることが可能となります。

### 5.2. 概要

CaoSQL プロバイダは、CaoSQL が持つ登録情報(CaoSQLConfig ツールで設定する情報)を元に、CaoSQL のインターフェースを取得し、それを内部で CAO のインターフェースと読み替えています。つまり、CaoSQL のコントローラ情報を一つの CaoController と見なすことができます。これにより、CaoSQL をカスケードしているかのようにシステムを構築することができます。(「図 5-1 CaoSQL のカスケード」参照)

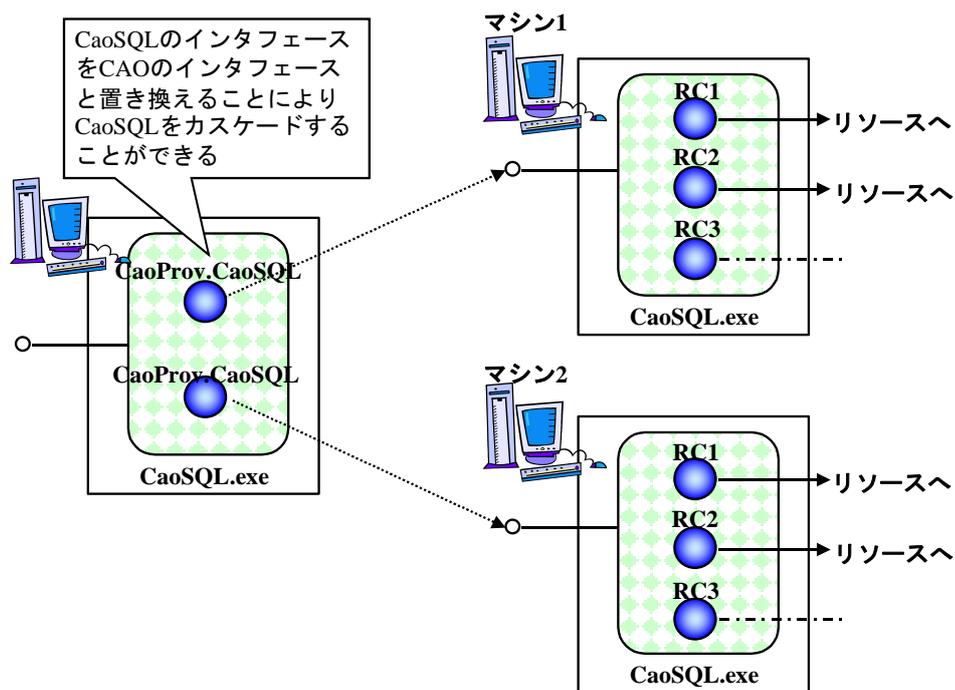


図 5-1 CaoSQL のカスケード

## 5.3. CaoProvCAOSQL.dll の使い方

### 5.3.1. はじめに

CaoProvCAOSQL.dll は CAO のプロバイダであり、基本的に CAO のインタフェースを介してアクセスします。このプロバイダはコントローラクラスと変数クラスが実装されていて、CAO から使用する際にもこの 2 つのクラスを使用することになります。

表 5-1 CaoProvCAOSQL.dll 情報

ファイル名	CaoProvCAOSQL.dll
ProgID	CaoProv.DNWA.CAOSQL
レジストリ登録	regsvr32 CaoProvCAOSQL.dll
レジストリ登録の抹消	regsvr32 /u CaoProvCAOSQL.dll

CaoSQL プロバイダで使用できる機能は、プロバイダテンプレートが提供する機能以外ではコントローラオブジェクトの取得、コントローラからコマンドの実行、変数オブジェクトの取得、変数の属性取得、変数の値の取得/設定等です。詳しくは表 5-4 を参照して下さい。

表 5-2 CaoSQL プロバイダ 利用可能メソッド/プロパティ一覧

オブジェクト名	メソッド/プロパティ名
CaoWorkspace	AddController
CaoController	AddVariable
	Execute
	get_Help
	get_VariableNames
CaoVariable	get_ID
	get_Value
	put_Value
	get_Attribute
	get_DateTime
	get_Help
	get_ID
	put_ID
get_Microsecond	

### 5.3.2. CaoWorkspace::AddController メソッド

CaoSQL プロバイダでは Controller オブジェクトの生成時に CaoSQLEngine 及び CaoSQLController オブジェクトを生成します。

表 5-3 CaoWorkspace::AddController のオプション文字列

オプション	説明						
Controller =<CaoSQL コントローラ名>	<p>CaoSQL に登録されているコントローラ名 (デフォルト:プロバイダのコントローラ名) 以下の予約名もコントローラ名として使用することができます。</p> <table border="1"> <thead> <tr> <th>コントローラ名</th> <th>説明</th> </tr> </thead> <tbody> <tr> <td>\$ENGINE\$</td> <td> <p>CaoController::get_VariableNames で, CaoSQL に登録されているコントローラ名の一覧を取得することができます。 CaoController::Execute で CaoSQLEngine::Execute を実行することが出来ます。 このコントローラ名で CaoController オブジェクトを作成した場合,get_VariableNames, Execute 以外は実行できません。</p> </td> </tr> <tr> <td>\$HISTORY\$</td> <td> <p>CaoController::Execute メソッドで, CaoSQL のヒストリーデータベースに対して SQL を実行することができます。 このコントローラ名で CaoController オブジェクトを作成した場合,Execute メソッド以外は実行できません。</p> </td> </tr> </tbody> </table>	コントローラ名	説明	\$ENGINE\$	<p>CaoController::get_VariableNames で, CaoSQL に登録されているコントローラ名の一覧を取得することができます。 CaoController::Execute で CaoSQLEngine::Execute を実行することが出来ます。 このコントローラ名で CaoController オブジェクトを作成した場合,get_VariableNames, Execute 以外は実行できません。</p>	\$HISTORY\$	<p>CaoController::Execute メソッドで, CaoSQL のヒストリーデータベースに対して SQL を実行することができます。 このコントローラ名で CaoController オブジェクトを作成した場合,Execute メソッド以外は実行できません。</p>
コントローラ名	説明						
\$ENGINE\$	<p>CaoController::get_VariableNames で, CaoSQL に登録されているコントローラ名の一覧を取得することができます。 CaoController::Execute で CaoSQLEngine::Execute を実行することが出来ます。 このコントローラ名で CaoController オブジェクトを作成した場合,get_VariableNames, Execute 以外は実行できません。</p>						
\$HISTORY\$	<p>CaoController::Execute メソッドで, CaoSQL のヒストリーデータベースに対して SQL を実行することができます。 このコントローラ名で CaoController オブジェクトを作成した場合,Execute メソッド以外は実行できません。</p>						
Machine=<マシン名>	<p>CaoSQL が起動している(又は起動したい)マシン名。 (デフォルト:プロバイダの実行マシン名)</p>						
ConvToArray=True/False	<p>ADO の Recordset オブジェクトから VARIANT 配列への変換を有効にします。(デフォルト:False) このオプションはコントローラ名に"\$HISTORY\$"を指定した場合にのみ有効になります。</p>						

```

AddController
(
    "<コントローラ名>",           // プロバイダのコントローラ名.
    "CaoProv. DNWA. CAOSQL",     // プロバイダ名. (固定)
    "[<マシン名>]",             // プロバイダの実行マシン名.
    "[[Controller=<CaoSQL コントローラ名>][, Machine=<マシン名>]]"
    // 登録されている CaoSQL コントローラ名, CaoSQL 実行マシン名.

```

)

(例)

```

AddController
(
    "CSQsCtrl1",           // プロバイダのコントローラ名に Ctrl1 を使用
    "CaoProv. DNWA. CAOSQL", // プロバイダ名. (固定)
    "",                   // ローカルマシンでプロバイダを実行 (in-process)
    "Controller=RC1, Machine=machine1"
    // 登録されているコントローラ RC1 を使用、“machine1”の CaoSQL に接続
)

```

### 5.3.3. CaoController::AddVariable メソッド

CaoSQL プロバイダでは Variable オブジェクトの生成時に CaoSQLItem オブジェクトを生成します。AddVariable で指定した変数名は CaoSQLItem の名前として使用されるので、取得したい CaoSQL アイテム名を入力します。

```

AddVariable
(
    "<CaoSQL 変数名>"           // プロバイダの変数名 = CaoSQLItem 名
)

```

### 5.3.4. CaoController::Execute メソッド

このメソッドで実行される内容は、AddController 時のコントローラ名によって異なります。各コントローラ名の毎の詳細な内容を以下に示します。

#### (1) \$ENGINE\$の場合

CaoSQLEngine の Execute メソッドを実行します。

引数に設定したコマンド文字列をそのままCaoSQLEngine::Executeに渡します。引数のパラメータ文字列は使用しません。

このメソッドの実行結果には、CaoSQLEngine::Execute の実行結果をそのまま返します。

ただし、下記のコマンドの場合はそれぞれの処理が実行されます。

コマンド名	引数	意味
\$CSQP_GET_COUNT\$	なし	CaoSQLController オブジェクト数(get_Count)

#### (2) \$HISTORY\$の場合

CaoSQLHistory の Execute メソッドを実行します。

引数に設定したコマンド文字列をそのまま CaoSQLHistory::Execute に渡します。引数のパラメータ文字列は使用しません。

このメソッドの実行結果は、AddController 時の ConvToArray オプションの値によって以下のようになります。

- ConvToArray = False  
CaoSQLHistory::Execute の実行結果を ADO の Recordset オブジェクトで返します。
- ConvToArray = True  
CaoSQLHistory::Execute の実行結果を VARIANT 配列で返します。  
VARIANT 配列の先頭要素には、実行結果のフィールド名リストを格納します。2番目以降の要素には、実行結果のレコードを順番に格納します。

(3) 上記2つ以外の場合

CaoSQLController の Execute メソッドを実行します。

引数に設定したコマンド文字列及びパラメータ文字列をそのまま CaoSQLController::Execute に渡します。

このメソッドの実行結果には、CaoSQLController::Execute の実行結果をそのまま返します。

ただし、下記のコマンドの場合はそれぞれの処理が実行されます。

コマンド名	引数	意味
\$CSQP_GET_COUNT\$	なし	CaoSQLItem オブジェクト数 (get_Count)
\$CSQP_GET_STATES\$	なし	CaoSQLController の状態 (get_State)

### 5.3.5. CaoController::get\_Help プロパティ

CaoSQL コントローラの Descripton プロパティを取得します。

### 5.3.6. CaoController::get\_ID プロパティ

CaoSQL コントローラの Index プロパティを取得します。

### 5.3.7. CaoController::get\_Tag プロパティ

CaoSQL コントローラの Tag プロパティを取得します。

### 5.3.8. CaoController::put\_Tag プロパティ

CaoSQL コントローラの Tag プロパティを設定します。

### 5.3.9. CaoController::get\_VariableNames プロパティ

コントローラ名が“\$ENGINE\$”の場合は、CaoSQL に登録されている CaoSQLController の一覧を取得します。

それ以外のコントローラ名の場合は、AddController で指定した CaoSQLController 配下の CaoSQLItem の一覧を取得します。

### 5.3.10. CaoVariable::get\_Attribute プロパティ

CaoSQL アイテムの属性値を取得します。取得する属性値は以下の通りです。

表 5-4 取得可能な属性値

属性値	意味
0	Read/Write
1	ReadOnly
2	WriteOnly

### 5.3.11. CaoVariable::get\_DateTime プロパティ

CaoSQL アイテムの更新日時を取得します。

### 5.3.12. CaoVariable::get\_Help プロパティ

CaoSQL アイテムの Descripton プロパティを取得します。

### 5.3.13. CaoVariable::get\_ID プロパティ

CaoSQL アイテムの ID を取得します。

### 5.3.14. CaoVariable::put\_ID プロパティ

CaoSQL アイテムの ID を設定します。

### 5.3.15. CaoVariable::get\_Microsecond プロパティ

CaoSQL アイテムの更新日時のマイクロ秒を取得します。

### 5.3.16. CaoVariable::get\_Tag プロパティ

CaoSQL アイテムの Tag を取得します。

### 5.3.17. CaoVariable::put\_Tag プロパティ

CaoSQL アイテムの Tag を設定します。

### 5.3.18. CaoVariable::get\_Value プロパティ

CaoSQL アイテムの値を取得します。

### 5.3.19. CaoVariable::put\_Value プロパティ

CaoSQL アイテムに値を設定します。

## 6. リンクビューア4

### 6.1. 概要

リンクビューアは、CaoSQLConfig で設定された LinkItem データを基にリンク状態を表示します。

また、リンクビューアは Microsoft Excel のアドインとして動作します。Excel のアドインの設定は 6.2 を参照して下さい。

1つのアイテムから複数にリンクしている状態でも、どのアイテムにリンクしているかすぐ把握でき、さらにリンクの数や CaoSQLConfig で設定された Description の表示なども行うことができます。

リンクビューアは、実行する際に作成されるシートが既に存在している場合、そのシートを削除し再作成します。同ファイルで別のリンク情報を取得したいときは、シートを別にコピーするなど準備しておく必要がありますので注意して下さい。

OUT	LINK	IN	DESCRIPTION
EMU Input		EMU Input	EMU入力(ラダーとの入出力用)
N7_00		NT_00	初期化完了
N7_01		NT_01	LEGOPICK LEGO取る
N7_02		NT_02	LEGOPICK パレット 動作
N7_03		NT_03	LEGOPICK ベルトへ
N7_04		NT_04	LEGO移動
N7_05		NT_05	LEGOキャッチ
N7_06		NT_06	LEGO持ち上げ
N7_07		NT_07	TABLEへ移動
N7_08		NT_08	LEGO置く
N7_09		NT_09	姿勢戻る
N7_10		NT_10	前方向き
N7_11		NT_11	検査
N7_12		NT_12	検査結果
N7_13		NT_13	1サイクル終了
N7_14		NT_14	検査中フラグ
1	EMU Output	EMU Output	EMU入力(ラダーとの入出力用)
N8_00		N8_00	初期化完了
N8_01		N8_01	LEGO取る
N8_02		N8_02	ベルトへ移動
N8_03		N8_03	LEGOをベルトへ置く
N8_04		N8_04	LEGO移動
N8_05		N8_05	LEGOキャッチ
N8_06		N8_06	LEGO持ち上げ
N8_07		N8_07	TABLEへ移動
N8_08		N8_08	LEGO置く
N8_09		N8_09	姿勢戻る
N8_10		N8_10	前方向き
N8_11		N8_11	検査
N8_12		N8_12	OKテーブルへ
N8_13		N8_13	NGテーブルへ
N8_14		N8_14	1サイクル終了
N8_15		N8_15	検査中フラグ
N8_16		N8_16	OKだけ通すためのフラグ
N8_17		N8_17	NGだけ通すためのフラグ
ERROR		ERROR	エラー
1	Error	Error	エラー値をラダーの入力へ
Random		Random	乱数
1	Random	Random	乱数
Trigger		Trigger	ラダーの出力をトリガとする
PATLITE		PATLITE	
RESET		RESET	
SET		SET	
1	POP	POP	POP外部出力
Data1		Data1	
Data2		Data2	

図 6-1 リンクビューアの全体画面

\*本ツールは、(財)機械振興協会 技術研究所が競輪の補助により実施した研究成果の一部を使用しています。また、本ツールはサポート対象外ですがソース公開されています。

## 6.2. アドインの追加

このリンクビューアを使用するには、Microsoft Excel のアドインとして動作させる必要があります。アドインを使用するには、アドインファイル(CaoSQL.xla)を登録しなければなりません。

Excel アドインは以下のフォルダに目的別に置くのが通例となっています。

- 全ユーザー共通 : [Office インストールディレクトリ]¥Office¥Library
- ユーザー別 : [ユーザー別 Profile ディレクトリ]¥Application Data¥Microsoft¥AddIns

例えば、Windows XP を使用しているユーザは、「C:¥Documents and Settings¥<ユーザ毎のフォルダ>¥Application Data¥Microsoft¥AddIns」の下にファイルを置きます。図 6-2 は Windows XP のフォルダ位置です。但し、環境<sup>5</sup>によっては、AddIns フォルダへのドライブパスが変わるので注意が必要です。

また、上記のフォルダに置かなくてもアドイン選択画面の[参照]でアドインファイルを指定することでもアドインファイルの登録ができます。



図 6-2 AddIns フォルダ(ユーザ別)

<sup>5</sup>今回の例は以下の環境で行っています。

OS: Windows XP, OS のインストール先ドライブ: Cドライブ, Office: Microsoft Office2000

以上の操作を行うことによって、Excel のアドイン(図 6-3)の一覧に「CaoSQL」が表示されます。

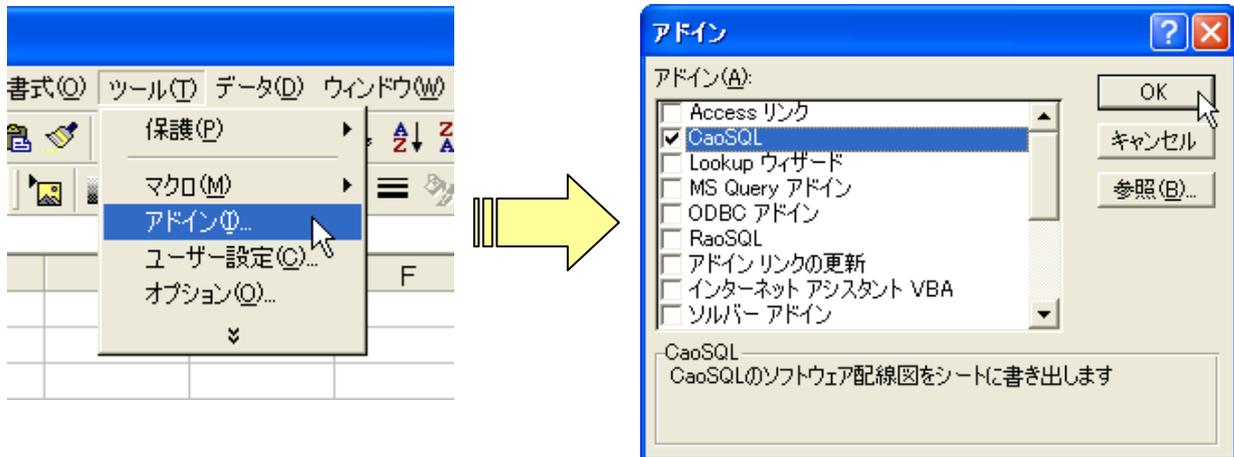


図 6-3 アドインの選択

アドイン一覧から「CaoSQL」にチェックして「OK」ボタンを選択する。Excel のメインメニューに「CaoSQL」が表示されていればアドインの追加は完了しています。(図 6-4)アドインの実行は次章で説明します。

リンクビューアの実行

CaoSQL をアドインすると、メニューに「CaoSQL」が表示されます。「CaoSQL」メニューの「LinkViewer 実行」を操作することにより、自動で

作成されるシートにリンクビューアが表示されます。

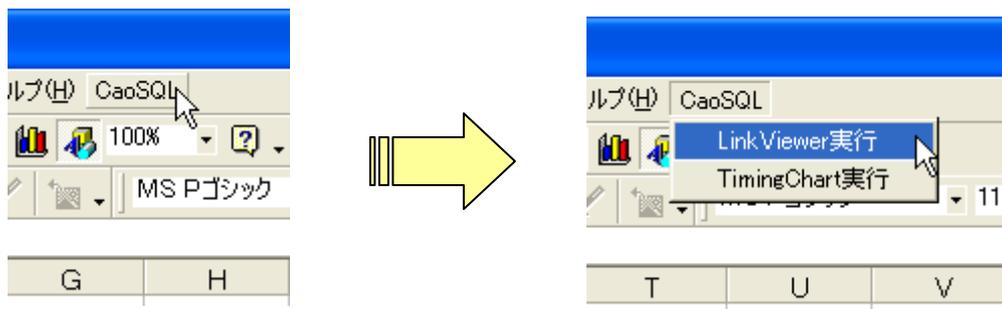


図 6-4 アドイン実行メニュー

リンクビューアを実行する際の注意事項として、「CaoSQL Connect」というシートが既にある場合、シートを再作成するために一度シートが削除されてしまう、ということがあります。

例えば、同ファイル上で複数の CaoSQL の Link 状態を示したシートが必要な場合、一度作成したシートの名前のままであると削除され再作成されるので、実行する事前に別シートへのコピーするか一度作成したシートの名前を変更するといったことが必要になります。まったく同じ Link 情報であれば、シートの保管は必要ありません。

リンクビューアを実行した結果は図 6-5 のようになります。

A	B	C	D	E	F
1	ソフトウェア配線図				
2					
3	1	EMU Input		EMU Input	EMU入力ラダーの入出力用)
4		N7_00		N7_00	初期比完了
5		N7_01		N7_01	LEGOPICK LEGO取る
6		N7_02		N7_02	LEGOPICK バレット動作
7		N7_03		N7_03	LEGOPICK ベルトへ
8		N7_04		N7_04	LEGO移動
9		N7_05		N7_05	LEGOキャッチ
10	1	N7_06		N7_06	LEGO持ち上げ
11		N7_07		N7_07	TABLEへ移動
12		N7_08		N7_08	LEGO置く
13		N7_09		N7_09	姿勢戻る
14		N7_10		N7_10	前方向き
15		N7_11		N7_11	検査
16		N7_12		N7_12	検査結果
17		N7_13		N7_13	1サイクル終了
18		N7_14		N7_14	検査中フラグ
19	3	EMU Output		EMU Output	EMU入力ラダーの入出力用)
20		N9_00		N9_00	初期比完了
21		N9_01		N9_01	LEGO取る
22		N9_02		N9_02	ベルトへ移動
23		N9_03		N9_03	LEGOベルトへ置く
24		N9_04		N9_04	LEGO移動
25		N9_05		N9_05	LEGOキャッチ
26		N9_06		N9_06	LEGO持ち上げ
27		N9_07		N9_07	TABLEへ移動
28		N9_08		N9_08	LEGO置く
29	3	N9_09		N9_09	姿勢戻る
30		N9_10		N9_10	前方向き
31		N9_11		N9_11	検査
32		N9_12		N9_12	OKテーブルへ
33		N9_13		N9_13	NGテーブルへ
34		N9_14		N9_14	1サイクル終了
35		N9_15		N9_15	検査中フラグ
36		N9_16		N9_16	OKだけ通すためのフラグ
37		N9_17		N9_17	NGだけを通すためのフラグ
38	2	ERROR		ERROR	エラー
39	1	Error		Error	エラー値をラダーの入力へ
40	1	Random		Random	乱数
41		Trigger		Trigger	ラダーの出力をトリガとする
42		PATLITE		PATLITE	
43		RESET		RESET	
44		SET		SET	
45	1	POP		POP	POP外部出力
46		Data1		Data1	
47		Data2		Data2	

図 6-5 リンクビューアの実行結果

## 6.3. リンクビューアの詳細

### 6.3.1. リンク先の数表示

ここでは、リンクビューアの表示の説明をします。図 6-6 は、先ほどの実行結果図 6-5 の一部分です。

リンクビューアは表の一番左側の列にリンク先が設定されている数を表示しています。

水色で塗りつぶされている行はコントローラを表しており、リンク先が設定されているアイテム数の合計が表示されています。図 6-6 の場合はアイテム「Error」と「Random」の 2 つがリンク先を指定しているため、コントローラ「ERROR」のリンク先情報セルには「2」が表示されます。

右のリンクされている側のセルには、リンクされている数などの情報表示はありません。

正常なリンク先が指定されているアイテムは青文字でアイテム名が表示されます。

リンク先が設定されている数を示しています

34		NB 14	NB 14	1サイクル終了
35		NB 15	NB 15	検査中フラグ
36		NB 16	NB 16	OKだけ流すためのフラグ
37		NB 17	NB 17	NGだけ流すためのフラグ
38	2	ERROR	ERROR	エラー
39	1	Error	Error	エラー値をラダーの入力へ
40	1	Random	Random	乱数
41		Trigger	Trigger	ラダーの出力をトリガとする
42		PATLITE	PATLITE	
43		RESET	RESET	
44		SET	SET	
45	1	POP	POP	POP外部出力
46		Data1	Data1	
47		Data2	Data2	

図 6-6 リンク先の数情報

### 6.3.2. リンク先の不明によるエラー表示

リンクビューアで設定されているリンク先が無効なアイテムの場合、リンクはエラーとなります。その場合、エラー表示は左側のリンク先の数を表示するセルに「！」が赤文字で表示されます。

複数リンクが設定されている場合でも、1つのリンクでも無効なアイテムがリンク先にある場合やリンク先が無い場合などは、同様にエラー表示になります。このときのアイテム名も赤文字で表示されます。

もし、リンク先が無い場合や無効な場合でリンクがエラーになっていたとしても、リンク先の合計を表示しているコントローラの表示は、エラー表示になっているリンクも含んで表示されます。

リンクエラーが出ていても合計はエラーも含まれます。

リンク先がエラーの場合の表示「！」で表示されます。

34		N9_14	N9_14	1サイクル終了
35		N9_15	N9_15	検査中フラグ
36		N9_16	N9_16	OKだけを流すためのフラグ
37		N9_17	N9_17	NGだけを流すためのフラグ
38	3	ERROR	ERROR	エラー
39	!	Error	Error	エラー値をラダーの入力へ
40	1	Random	Random	乱数
41		Trigger	Trigger	ラダーの出力をトリガとする
42		PATLITE	PATLITE	
43		RESET	RESET	
44		SET	SET	
45	1	POP	POP	POP外部出力
46		Data1	Data1	
47		Data2	Data2	

図 6-7 リンク先の不明によるエラー

## 7. タイミングチャート6

### 7.1. 概要

タイミングチャートは、CaoSQL で記録したデータベースを基に、チャートを描くツールです。

タイミングチャートは Microsoft Excel のアドインとして動作し、Excel のシートにチャートが表示されます。

チャートに表示できる CaoSQLItem はユーザが任意に設定することができます。また、チャートに表示できるデータ時間は、ユーザが任意に設定した時間から 10 秒、20 秒、30 秒間のいずれかのデータです。

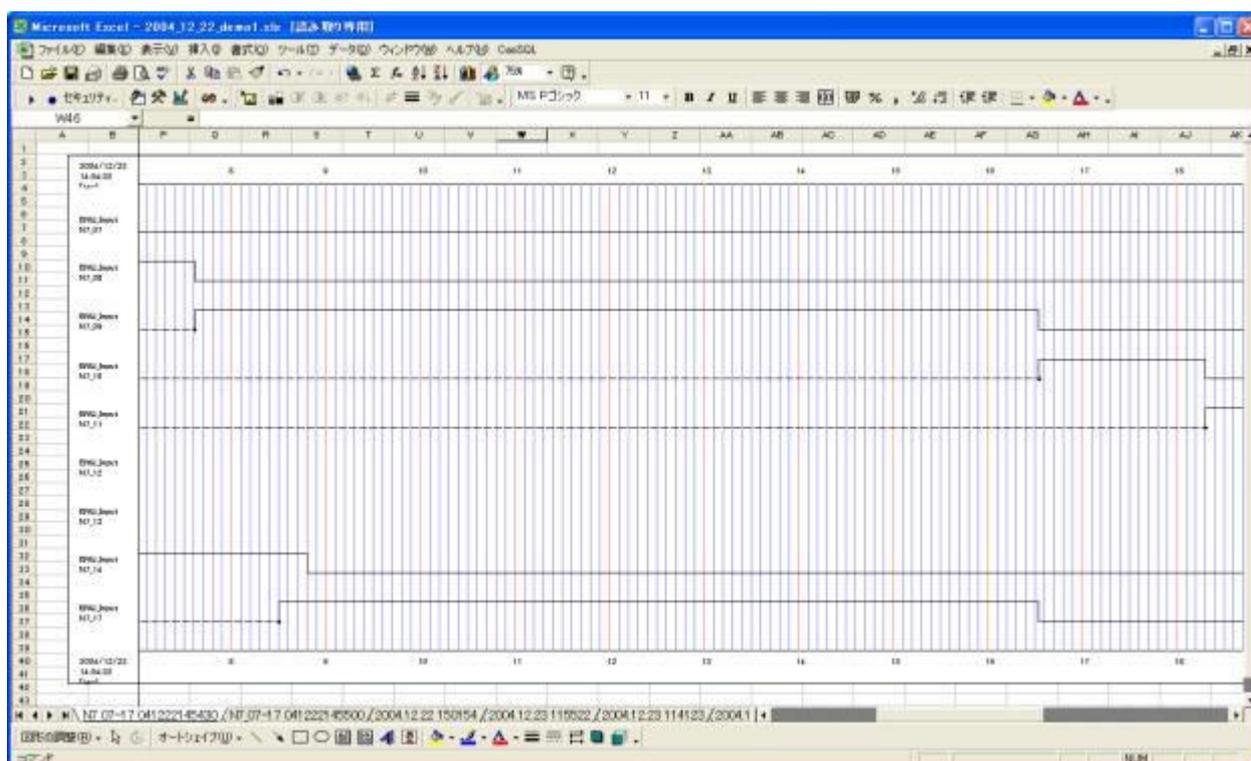


図 7-1 タイミングチャートの全体画面

<sup>6</sup>本ツールは、(財)機械振興協会 技術研究所が競輪の補助により実施した研究成果の一部を使用しています。また、本ツールはサポート対象外ですがソース公開されています。

## 7.2. アドインの追加

タイミングチャートを使用するには、まず Microsoft Excel のアドイン登録をする必要があります。アドインの追加方法は、リンクビューアと同じなので「6.2 アドインの追加」を参照してください。

## 7.3. タイミングチャートの使用方法

アドインを追加すると、図 7-2 のようなメニューが追加されます。そのメニューの「TimingChart 実行」を選択します。

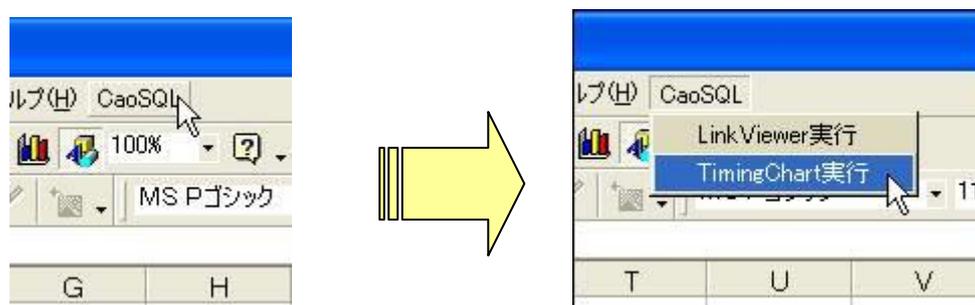


図 7-2 アドイン追加後のメニュー

メニューから実行されると、タイミングチャートを描くのに必要な設定を行うウィザードが起動します。(図 7-3)

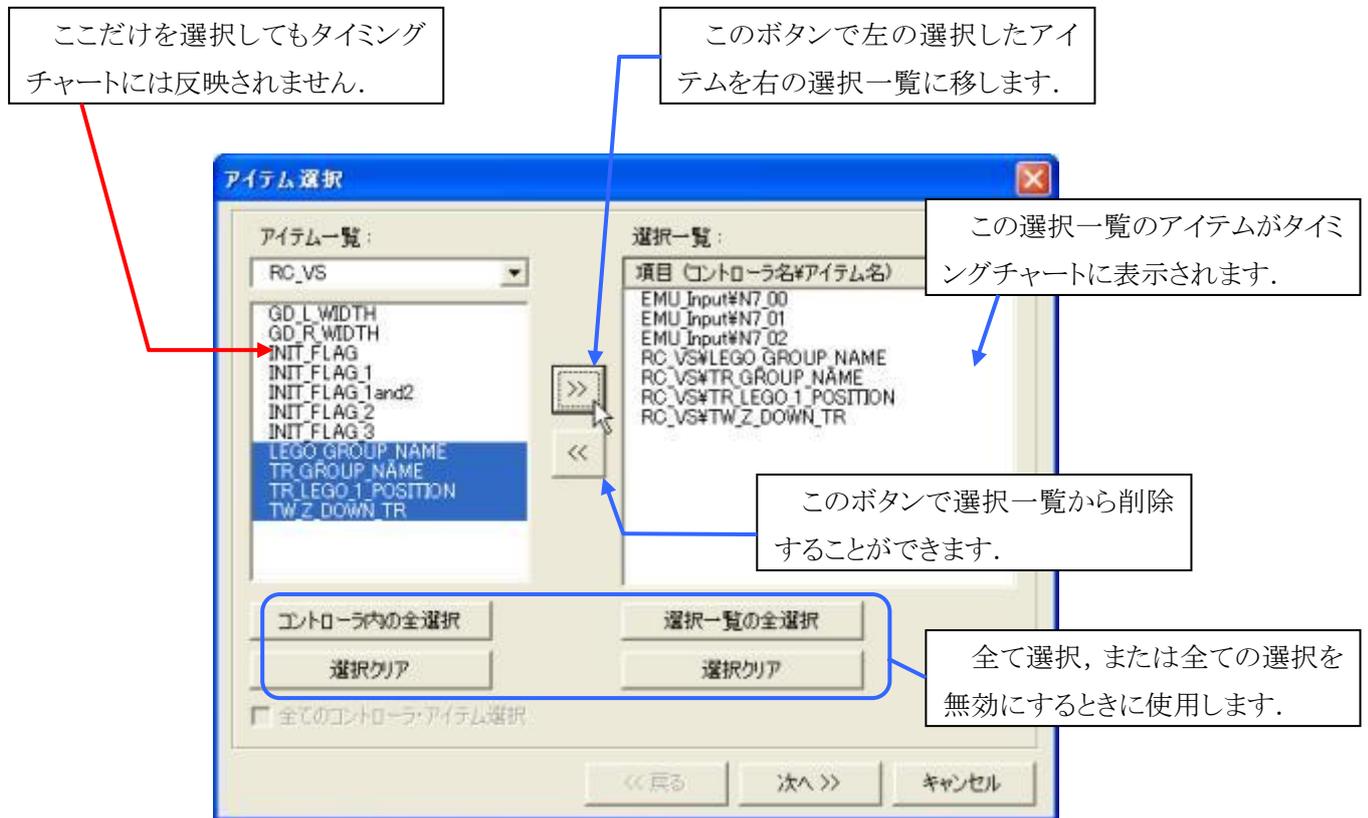


図 7-3 設定ウィザードのアイテム選択画面

まず、設定の最初にアイテム一覧の方から追加したいアイテムを選択します。この選択時には複数選択することもできます。ここで選択するだけではタイミングチャートに反映されないで注意が必要です。これらの選択を有効にするには、中央にある「>>」ボタンを押すと選択一覧にアイテムが追加されるので、この一覧のアイテムがタイミングチャートに反映されます。逆に選択一覧からアイテムを外したい場合は、選択一覧から外したいアイテムを選択して「<<」ボタンを押すことで外すことができます。選択一覧のアイテムは最大で 20 個まで追加できます。

アイテムの設定が完了したら、「次へ>>」を選択します。

次に、タイミングチャートを表示する時間を指定します。(図 7-4)



図 7-4 タイミングチャートを表示する時間の設定画面

ここで指定した時刻から、10 秒、20 秒、30 秒のいずれかの経過時間後のデータをタイミングチャートに起こします。不正な時刻を指定すると警告メッセージが表示されます。時間設定が完了したら「次へ>>」を選択します。

タイミングチャートを実行すると、シートが新規作成されます。新規作成されるシート名はここで指定することができます。デフォルトで指定されているシート名は、ウィザードの前ページで指定した時刻を文字列化した名前になっています。例えば、図 7-4 の場合は指定時刻が“2004 年 12 月 27 日 14 時 3 分 7 秒”なので、シート名は“2004.12.27 140307”となります。当然ユーザの任意の名前を指定することもできます。既にあるシート名を指定した場合、Microsoft Excel のデフォルトシート名が使用されます。

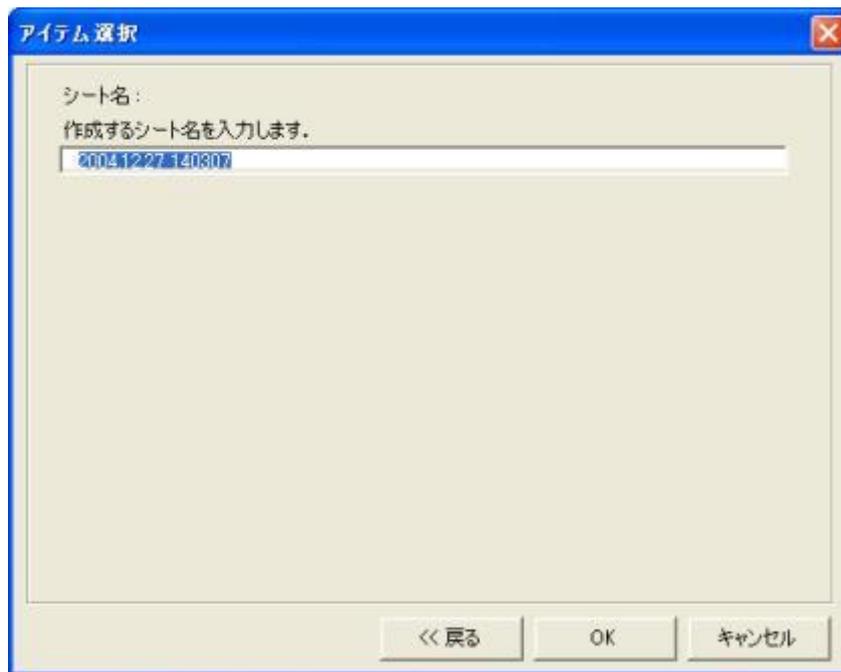


図 7-5 シート名の設定画面

全ての設定が完了したら、「OK」ボタンを選択します。アイテム数によっては、タイミングチャートが表示されるのに若干時間がかかる場合もあります。

#### 7.4. タイミングチャートの詳細

タイミングチャートは CaoSQL の History 機能を使用しています。CaoSQL が記録したデータベースを基にタイミングチャートを描きます。CaoSQLConfig の設定によりますが、CaoSQL は基本的にアイテムの値が変化した時にデータベースに書き込まれます。タイミングチャートは 0 または ≠0 で変化として書き出します。但し、最初の値は指定時刻より前のデータが 0 であったのか ≠0 だったのか、または Empty 値だったのか判定不可能なため、不定値として点線で表示します。また、アイテム名項目はウィンドウ枠を縦に固定していますので、スクロールしても時間とアイテム名はスクロールしません。

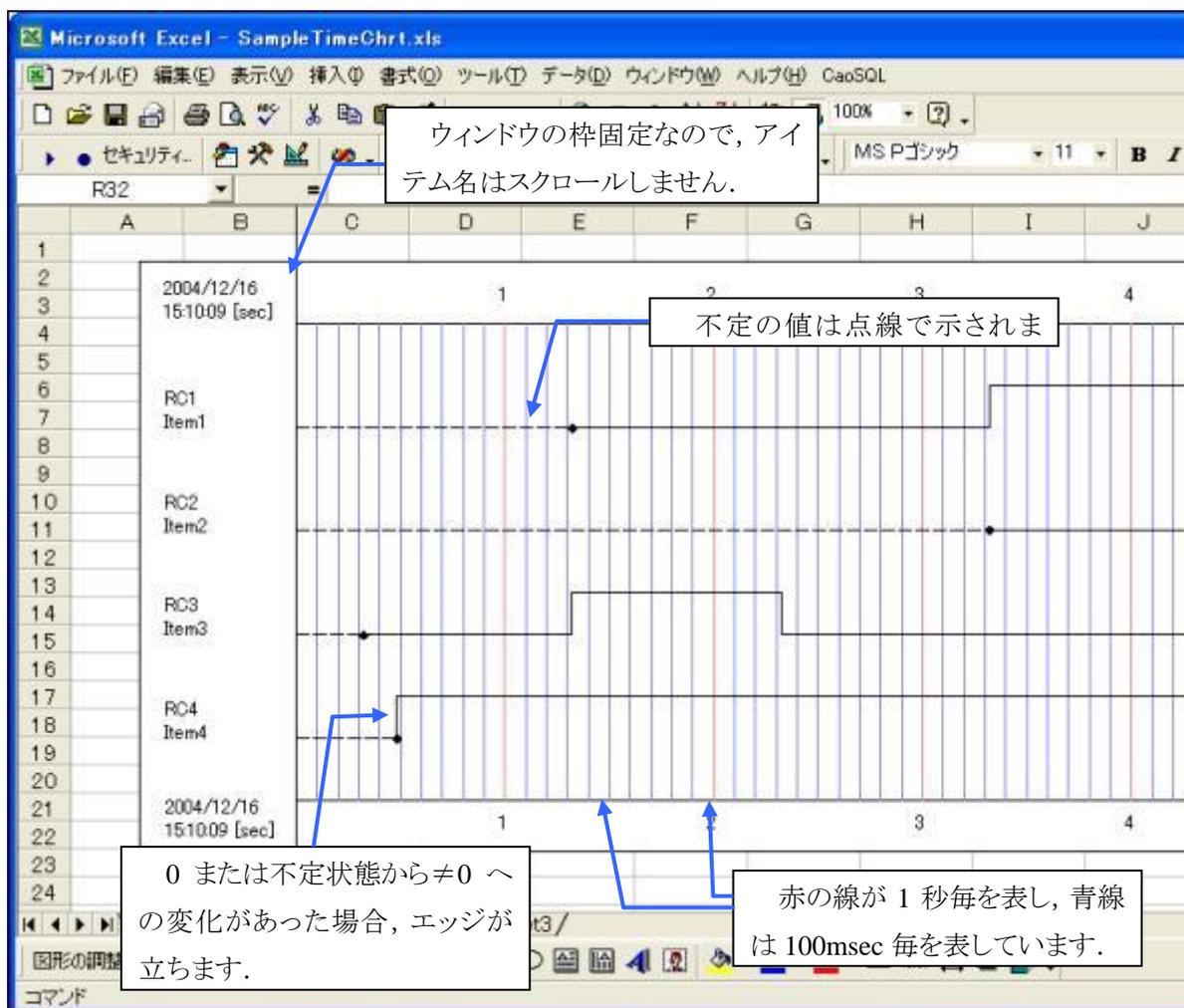


図 7-6 タイミングチャートの画面

## 8. jCaoSQL

### 8.1. 概要

jCaoSQL<sup>7</sup>は java から CaoSQL のコントロールをするためのフレームワークであり、CaoSQL のインタフェースを介して、ロボットコントローラなどのリソースにアクセスすることができます。現在のところ、一部制限はありますが CaoSQL の機能は全て java から使用することが可能です。

本章は、jCaoSQL を使用した java の CaoSQL クライアントアプリケーションの作成手順ですので、CaoSQL インタフェースの詳細は「[CaoSQL ユーザーズガイド](#)」を参照して下さい。

### 8.2. クライアント開発環境セットアップ

ここではjCaoSQLを利用したクライアント開発の方法について触れます。クライアント開発ではjCaoSQLプロジェクト内の以下のファイルを使用します。

表 8-1 jCaoSQL の使用ファイル

ファイル名	概要
	設定方法
jCaoSQL.jar	jCaoSQL の java バイナリパッケージ
	ClassPath に通します。
jCaoSQL.dll	jCaoSQL のネイティブコード
	クライアントから参照できる場所に置く。または、環境変数の%PATH%に場所を登録します。
jcaosql_log.config	ネイティブコードのログ設定情報
	必須ではありませんが、ネイティブコードのログ情報を必要とする場合は、アプリケーション実行フォルダに置きます。

具体的なクライアントの開発方法については、以下のドキュメントと、パッケージに追加されている sample フォルダを参考にして下さい。

### 8.3. サンプル解説

#### 8.3.1. インスタンスの生成と破棄

CaoSQLEngine, CaoSQLController, CaoSQLItem オブジェクトは、それぞれ

CaoSQLEngine#createInstance, CaoSQLEngine#getController, CaoSQLController#getItem がファクトリメソッドになっており、この際に対応する COM のインスタンスが生成されます。従って、これらのオブジェクトを破棄する際には必ず release をコールする必要があります。java ではデストラクタの概念が無いので、クライアント

<sup>7</sup>本ツールはサポート対象外です。

アプリケーションで明示的にコールして下さい。

このサンプルは `sample¥sample1` フォルダにあり, `sample` フォルダから以下のコマンドでサンプルの実行を確認することができます。

```
> ant sample1
```

### 8.3.2. VARIANT 型の対応

jCaoSQL では VARIANT クラスを `java.lang.Object` クラスにマッピングしています。ここでは, java クライアント側での VT 型の判別サンプルを示します。

java では `instanceof` 演算子を使うことで, オブジェクトの型を持つかどうか判定することが可能です。例えば, VT\_I4 型の VARIANT であれば `java.lang.Integer` クラスにマッピングされていますので, 以下のような処理で VT\_I4 であることの判別が可能です。

```
if (value instanceof Integer) {  
    // VT_I4 型変数  
}
```

また, 配列型に関してはオブジェクトから `java.lang.Class` を取得し, この Class オブジェクトのメソッド `isArray` を呼ぶことで, 配列型かどうかの判別が可能です。

このサンプルは `sample¥sample2` フォルダにあり, `sample` フォルダから以下のコマンドでサンプルの実行を確認することができます。

```
> ant sample2
```

### 8.3.3. イベントシンの作成

jCaoSQL ではサンプルのようにイベント受信が可能です, CaoSQL は非同期的にイベントを発生させるので, java オブジェクト側で同期を取りたい場合は `jp.co.denso.fa.CaoSQLControllerListener` の実装時に `OnChangeListener` メソッド, `OnChangeState` メソッド, あるいはメソッド内で適宜 `synchronized` を宣言し, 同期を図って下さい。

また, `OnChangeListener` イベントの引数で渡された `CaoSQLItem` オブジェクトは 8.3.1 と同じく破棄(明示的な `release` のコール)しなければなりません。

このサンプルは `sample¥sample3` フォルダにあり, `sample` フォルダから以下のコマンドでサンプルの実行を確認することができます。

```
> ant sample3
```

### 8.3.4. カスタムクラスの実装

jCaoSQL を作成している際に, `CaoSQLJava` オブジェクトで独自の機能を実装したカスタムクラスを作成した方が設計や実装上で便利な場合があります。こういった場合は `jaosql.properties` にカスタムクラスを定義し,

既存の CaoSQLJava オブジェクトの派生クラスを作成することで実現が可能です。

以下にその方法を示します。

表 8-2 jcaosql.properties のキー一覧

キー名	意味
caosql.engine.iid	CaoSQLEngine のインタフェース ID(通常は変更不要)
caosql.engine.progid	CaoSQLEngine のプログラム ID(通常は変更不要)
caosql.engine.class	ファクトリが作成する CaoSQLEngine クラス デフォルトは“jp.co.denso.fa.jcaosql.CaoSQLEngine”
caosql.controller.class	ファクトリが作成しようとする CaoSQLController クラス デフォルトは“jp.co.denso.fa.jcaosql.CaoSQLController”
caosql.item.class	ファクトリが作成しようとする CaoSQLItem クラス デフォルトは“jp.co.denso.fa.jcaosql.CaoSQLItem”

例えば、CaoSQLItem を基底クラスとする com.example.MyCaoSQLItem クラスを使用したい場合は jcaosql.properties という名前のファイルを作成し、キー<jcaosql.item.class>に、値<com.example.MyCaoSQL>を書き、実行時のルートにこの properties ファイルを配置します。この処置でファクトリクラスは CaoSQLItem の作成時に com.example.MyCaoSQLItem クラスを作成するようになります。

このサンプルは sample¥sample4 フォルダにあり、sample フォルダから以下のコマンドでサンプルの実行を確認することができます。

```
> ant sample4
```

### 8.3.5. アイテムの動的追加, 削除

jCaoSQL は CaoSQL と同じく動的にアイテムを追加する事が可能です。削除可能なアイテムが動的に追加したアイテムのみであるのも CaoSQL と同じです。

このサンプルは sample¥sample5 フォルダにあり、sample フォルダから以下のコマンドでサンプルの実行を確認することができます。

```
> ant sample5
```