
RTA-TRACE

レポートマークアップランゲージ リファレンスガイド

著作権について

本書のデータを LiveDevices Ltd. からの通知なしに変更しないでください。LiveDevices Ltd. は、本書に関してこれ以外は一切の責任を負いかねます。本書に記載されているソフトウェアは、お客様が一般ライセンス契約あるいは単一ライセンスをお持ちの場合に限り使用できます。ご利用および複写はその契約で明記されている場合に限り、認められます。

本書のいかなる部分も、LiveDevices Ltd. からの書面による許可を得ずに、複写、転載、伝送、検索システムに格納、あるいは他言語に翻訳することは禁じられています。

© **Copyright 2004** LiveDevices Ltd.

本書で使用する製品名および名称は、各社の（登録）商標あるいはブランドです。

Document TD00010-002

目次

1	本書について	5
1.1	本書の対象ユーザー	5
1.2	表記上の規約	5
2	使用法	6
2.1	カスタムレポートの生成	6
2.1.1	レポートテンプレートを作成する	6
2.1.2	レポートテンプレートを使用する	7
2.1.3	レポートを表示する	7
2.1.4	例 1 (HTML)	7
2.1.5	例 2 (HTML)	8
2.1.6	例 3 (CSV)	8
2.2	エラー	9
3	リファレンス	10
3.1	テキスト	10
3.2	コメント	11
3.3	コマンドトークン	12
3.3.1	font	12
3.3.2	fmt	13
3.3.3	repeat	14
3.3.4	for	15
3.3.5	do	16
3.3.6	if	17
3.3.7	nl	17

3.4	変数アクセス	17
3.5	オブジェクト属性と統計値の参照	18
3.6	オブジェクトのシノニム	19
4	お問い合わせ先	21
	索引	23

1 本書について

RTA-TRACE は組込みシステム用のソフトウェアロジックアナライザです。アプリケーションと組み合わせて使用することにより、システムのデバッグやテストに役立つさまざまなサービスを利用できます。中でも特に優れた機能として、量産用にビルドされたアプリケーションソフトウェアについて、ランタイムにシステム内で起こっている事象を正確に把握することができます。

RTA-TRACE クライアント用レポートプラグインを使うと、ターゲットアプリケーションから得られるトレースデータを元にカスタムレポートを生成できます。本書では、これらのレポートのカスタマイズを行う際に利用できる RTA-TRACE 「レポートマークアップランゲージ」(RTA-TRACE Report Mark-up Language) について説明します。

1.1 本書の対象ユーザー

本書は、RTA-TRACE で得られたデータを元にしてカスタムレポートを作成しようとするユーザーを対象としています。また、『RTA-OSEK 入門ガイド』や『RTA-OSEK リファレンスガイド』に従って RTA-TRACE が正しくセットアップされ、稼働していることも前提条件となります。

1.2 表記上の規約

重要：このように表記されている注記には、ユーザーが知っておく必要のある重要な情報が記載されています。内容をよく読み、記載されているすべての指示に必ず従ってください。

移植性：このように表記されている注記では、RTA-OSEK コンポーネントが実行されるプロセッサ上で実行できるコードを作成する場合に知っておく必要がある事柄について説明されています。

本書では、プログラムコード、ヘッダファイル名、C のデータ型名、C 関数および API 関数名はすべてクーリエ体 (courier) で表記されています。オブジェクトの名前も、プログラマに公開され次第やはりクーリエ体で表記されます。たとえば、Task1 という名前のタスクは、`Task1` という名前のタスクハンドルとして表記されます。

GUI エlementとのインタラクションについての記述では、Elementのキャプションは**ボールド体 (bold)** で表記されています。また、メニューなどの階層的なナビゲーションは矢印でレベルを区切り、たとえば、「メニューコマンド **Edit** → **Select All** を選択します。」、または「メニューから **Edit** → **Select All** を選択します。」のように表記されています。

また PDF 文書において、索引、および他の部分を参照する箇所（例：「第 3 章を参照してください」の部分）については、その参照先へのリンクが設けられているので、必要な参照箇所を素早く見つけることができます。

2 使用法

レポートは「レポートテンプレート」に基づいて作成され、ファイルに書き込まれます。ここでは、いくつかのサンプルレポートを例に、レポートの生成方法について説明します。利用可能なテンプレートエレメントについては、第3章で詳しく説明します。

カスタムレポートを生成するためには、追加のライセンスが必要です。詳しくは ETAS までお問い合わせください。

2.1 カスタムレポートの生成

カスタムレポートを生成するには、まずレポートテンプレートを作成する必要があります。レポートテンプレートにより、システムオブジェクトからさまざまなパラメータを抽出して固定的なテキスト文字列と適宜に組み合わせてレポート出力を行うことができます。

2.1.1 レポートテンプレートを作成する

レポートテンプレートはシステムオブジェクトからプログラマティックに情報を抽出して出力ファイルを生成するためのものです。まず簡単な例から説明します。

下のテンプレートは、オブジェクトのタイプ、名前、および識別子をリスト出力する処理を、システム内の全オブジェクトについて繰り返します。

```
Objects in this system:
$n1$
$n1$
$repeat:ob=all$
@ob.Type@ "\t" @ob.Name@ "\t" @ob.ID@
$n1$
$~repeat$
```

出力結果は以下のようになります（内容はシステムコンフィギュレーションにより異なります）。

```
Objects in this system:

ErrorHook65533
Cat2 ISRSysSystemISR16
TasktskLeader9
Processtask_tskLeader10
TasktskInterfere7
Std resourcerscContention2
Std resourceRES_SCHEDULER1
CounterSystemTimer11
AlarmalmLeader13
...
TracepointinterfereValue32771
IntervalinterfereRunTime40965
Background(Unallocated) 65531
OS activity65530
```

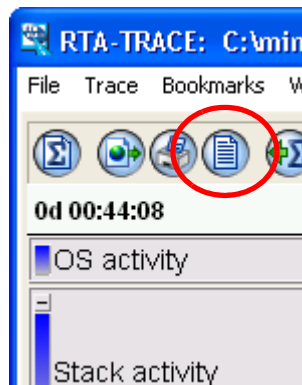
どのようなシステムについても、上の例のようなレポートテンプレートを複数適用できます。テンプレートは再利用できるので、複数のシステムについて共通の情報を抽出することもできます。

レポートテンプレートは、プレーンな ASCII テキストの形で保存されていれば、任意のテキストエディタで生成できます。レポートの最終的なフォーマットは任意に変更してください。2.1.4 項以降で HTML ファイル（ウェブブラウザ用）を作成する例と CSV（Comma Separated Value）ファイル（スプレッドシートで表示／分析用）を作成する例を紹介します。

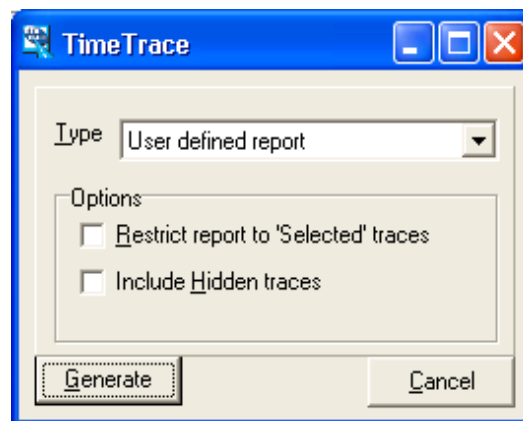
2.1.2 レポートテンプレートを使用する

テンプレートを用いてのレポート生成は、RTA-TRACE TimeTrace クライアントから以下の手順で行ってください。

1. RTA-TRACE クライアントが稼働していてトレースデータ（実データまたはファイルに保存されたデータ）が表示されている状態で、ツールバーの 'Generate Report' ボタンをクリックします。



2. すると下のダイアログボックスが表示されるので、'User defined report'（ユーザー定義のレポート）を選択します（下図参照）。



3. ファイル選択ダイアログボックスが表示され、レポートテンプレートが格納されているファイルを選択するように要求されます。適切なテンプレートを選択すると、別のファイル選択ダイアログボックスが表示され、生成されるレポートのファイル名とロケーションを選択するように要求されます。

これでレポートが生成されます。生成されたレポートは、ファイルのタイプに適したビューアで表示できます。

2.1.3 レポートを表示する

レポートに適したビューアは、生成されるレポートのタイプに応じて必然的に決まります。一般に、レポートは任意のテキストエディタで表示できますが、別のビューア（たとえば、CSV ファイルはスプレッドシート、HTML ファイルはウェブブラウザ）に表示した方が見やすい場合があります。

2.1.4 例 1 (HTML)

この例は、2.1.1 項で紹介したプレーンテキストの例に HTML タグを付けたものです。

```
<HTML>
<HEADER>
<TITLE>Objects in this system</TITLE>
```

```

</HEADER>
<BODY>
<H1>Objects in this system</H1>
$repeat:ob=all$
<P> @ob.Type@ "\t" @ob.Name@ "\t" @ob.ID@ </P>
$nl$
$~repeat$
</BODY>
</HTML>

```

上記の例では、表示に必要な HTML ヘッダが挿入されています。
必要に応じてこの情報をテーブル形式で出力したり、複雑なフォーマットを行うこともできます。

2.1.5 例 2 (HTML)

この例では、条件文を用いて、生成されるレポートの内容を制御する方法を示します。この場合もシステムの各オブジェクトが出力されますが、ここでは、ID が 32767 より大きいオブジェクトの名前はボールド体でハイライトされます。

生成されたレポートはテーブル形式で表示されます。

```

<HTML><HEADER>
<TITLE>Objects in this system</TITLE>
</HEADER>
<BODY>
<H1>Objects in the system @System@</H1>
<TABLE>
$nl$
$repeat:ob=all$
<TR><TD>
$if:@ob.ID@,gt,32767$<B>$~if$ // start BOLD
@ob.Type@
$if:@ob.ID@,gt,32767$</B>$~if$ // end BOLD
</TD>
<TD> @ob.Name@ </TD>
<TD> @ob.ID@ </TD>
</TR>
$nl$
$~repeat$
$nl$
</TABLE></BODY></HTML>

```

2.1.6 例 3 (CSV)

RTA-TRACEにより取得されたトレースデータを統計プラグインにより分析することができ、この統計データをレポートに取得することができます。

下の例は、システム内のタスクについての主要な統計値を出力するものです。生成されたファイルはスプレッドシートにインポートすることができ、さらに、スプレッドシートプログラムがグラフプロット機能をサポートしていれば、それをグラフとしてプロットすることもできます。

下のテンプレートは、CPU 使用率のテーブルを生成します。

```

$fmt:ticks$
$repeat:ob=tasks$
"@ob.Name@ , @ob.Utilization@"
$nl$
$~repeat$

```



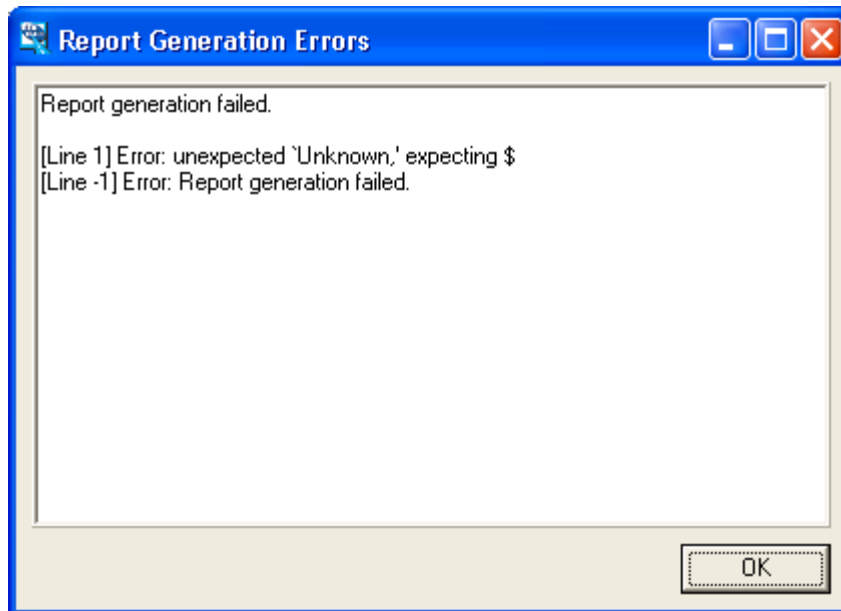
```
$repeat:ob=isrs$
"@ob.Name@ , @ob.Utilization@"
$n1$
$~repeat$
$n1$
```

これにより、スプレッドシートにインポートするのに適した CSV ファイルが生成されます。その後、測定期間内の「未知の」時間を計算する必要がありますが、これは、使用率の合計を 100 から引くだけで算出できます。(例: Microsoft Excel の場合、この計算は '=100-SUM(B1:Bn)' となります。式中の n はグラフ内のオブジェクト数です。)

使用率のようにあらかじめシステムに組み込まれている統計値を使うことはもちろん、ユーザー独自の統計値を TimeTrace プラグイン内で定義して、同じ方法で参照することができます。

2.2 エラー

レポートテンプレートに誤りがあると、Report Generation 機能は下図のようなエラーダイアログを表示します。



エラーメッセージには、エラーが発生した行が示されます。上図の行 '-1' は、特定の行だけに関係するエラーではないことを示しています。

3 リファレンス

カスタムレポートは以下のエレメントで構成されます。

- テキスト
- コメント
- コマンドトークン
- 変数の参照
- オブジェクトの参照

以降の部分では、これらのエレメントについて詳しく説明します。

3.1 テキスト

このプラグインは引用テキスト(前後に"が付いているテキスト)と非引用テキストをサポートしています。コマンドトークン(2つの\$記号で囲まれた文字 - 下記参照)は、引用テキスト内では機能が抑制され、通常のテキストとして扱われます。

注記

変数の参照とオブジェクトの参照(2つの@記号で囲まれた文字 - 下記参照)は、引用テキスト内でも非引用テキスト内でも参照として解釈されます。

引用テキスト内でも、ニューライン、キャリッジリターン、タブは有効です。

引用テキスト内では、'@'、'" および \' (\\$) の文字の前に \' (\\$) の文字を挿入することにより、これらの文字を通常のテキストとして扱うことができます。つまり、引用テキスト内ではオブジェクトの属性と統計値への参照が抑制され、また、引用記号とバックスラッシュ文字 (\\$ 文字) を引用テキストの文中に含めることができます。

またさらに、引用テキスト内で C 言語形式の 8 進エスケープシーケンスを使用することにより、任意の 8 ビット文字を挿入できます。

例 1 引用テキストと非引用テキストの比較。以下の記述により

```
This text
" is followed by "
more text
$nl$
"Command tokens such as $nl$ are
ignored in quoted text"
```

以下のように出力されます。

```
This text is followed by more text.
Command tokens such as $nl$ are
ignored in quoted text
```

例 2 特殊文字の抑制。以下の記述により

```
This text
" \"is followed by\" "
This other text
```

以下のように出力されます。

```
This text "is followed by" This other text.
```

例 3 オブジェクトへの参照。以下の記述により

```
The time is now @now@.  
$nl$  
"The value of \@now\@ is @now@."
```

以下のように出力されます。

```
The time is now 12/01/2004 10:57:57.  
The value of @now@ is 12/01/2004 10:57:57.
```

例 4 エスケープシーケンスによる任意の文字の挿入。以下の記述により

```
" \101 \102 \103 \n \104 \105 \106 "
```

以下のように出力されます。

```
A B C  
D E F
```

3.2 コメント

生成されるレポートには表示されないコメントを、レポートテンプレートに挿入することができます。コメントは//の文字で始め、ニューライン文字で終わります。

例 以下の記述により

```
This should /app//remove this  
ear/.
```

以下のように出力されます。

```
This should /appear/.
```

3.3 コマンドトークン

2つの\$の間に記述されるコマンドトークン（例：`$n1$`）により、レポートの出力制御を行うことができます。たとえば、システム内のすべてのタスクにわたる反復（`$repeat:...$`を使用します）や、出力に基づく出力内容などの決定（`$if:...$`を使用します）を行うことができます。

3.3.1 font

注記

このトークンは、できるだけ使用しないことをお勧めします。一般に、生成される内容と等価のテキストをそのまま使用の方がよいでしょう。

シンタックス

開始：`$font:string$`

終了：`$~font$`

説明

`$font:...$` トークンはテキスト `<string>` を出力ストリームに挿入します。

`$~font$` トークンは、`</string>` というテキストを出力ストリームに挿入します。

これらのコマンドトークンを使用して、HTML で使用されるさまざまなタグのペア（`<p>` と `</p>`、`<h1>` `</h1>` など）を挿入することができます。

`$font:...$` コマンドトークンは、ネスティングが可能です。

例

以下の記述により

```
$font:p$A paragraph with
```

```
$font:i$ italic$~font$ text.$~font$
```

以下のように出力されます。

```
<p>A paragraph with <i>italic</i>
text.</p>
```

シンタックス	<code>\$fmt:style\$</code>
説明	<p><code>\$fmt:...\$</code> トークンは後続の統計値の出力形式を変更します。デフォルトの出力形式は <code>str</code> です。</p> <p>有効な <code>style</code> の値は以下のとおりです。</p> <ul style="list-style-type: none"> • <code>ms</code> – ミリ秒の単位で出力されます（単位は表示されません）。 • <code>ticks</code> – チックの単位で出力されます（単位は表示されません）。 • <code>str</code> – 文字列として出力されます（適切な単位が表示されます）。 <p><code>style</code> の値が無効の場合にはデフォルトが適用されます。</p> <p>このトークンは、1つのレポート内で必要に応じて何回も繰り返して使用できます。</p>
例	<p>以下の抜粋部分は、ユーザーオブジェクト A についてそれぞれ異なるフォーマットタイプを生成します（3.5 項の、統計値の参照についての説明を参照してください）。</p> <pre>"default: time = @A.Time Stat 1@\n" \$fmt:ms\$ "ms : time = @A.Time Stat 1@\n" \$fmt:str\$ "str : time = @A.Time Stat 1@\n" \$fmt:ticks\$ "ticks : time = @A.Time Stat 1@\n" \$fmt:invalid\$ "invalid: time = @A.Time Stat 1@\n"</pre> <p>出力は以下のようになります。</p> <pre>default: time = 2ms 340us 100ns ms : time = 2.340100 str : time = 2ms 340us 100ns ticks : time = 23401 invalid: time = 2ms 340us 100ns</pre>

シンタックス

開始: `$repeat:var=name$`
 終了: `$~repeat$`

説明

`$repeat:...$` タグは、以下を *name* の値としてサポートしています。

- `tasks` と `task` – すべてのタスク (OSEK タイプのタスクと協調タスクを含みます。)
- `isrs` と `isr` – すべての ISR (Cat0、Cat1、Cat2 を含みます。)
- `resources` と `resource` – すべてのリソース (標準リソースとリンクリソース)
- `std_resources` と `std_resource` – 標準リソースのみ
- `intervals` と `interval` – インターバル
- `processes` と `process` – プロセス
- `tracepoints` と `tracepoint` – すべてのトレースポイント
- `profiles` と `profile` – すべてのタスク / ISR プロファイル
- `group` と `groups` – レポートされるすべてのグループ
- `all` – すべてのオブジェクト

`$repeat:...$` のループの処理では、カプセル化されているトークンはオブジェクトディクショナリ内の *name* にマッチする各項目につき 1 回解釈されます。

repeat ループが解釈される時には、変数 *var* にはマッチする各オブジェクトのオブジェクトタイプと名前が設定されます。このようにして作成される変数は、レポート内において、通常の変数アクセスメカニズムによってアクセスできます。

`$repeat:...$` ループはネスティングが可能です。

例

以下の記述により、システムの各オブジェクトのタイプのリストが作成されます。

```
$repeat:obj=all$
"Object is of type '@obj.Type@'\n"
$~repeat$
```

出力は下のようになります。

```
...
Object is of type 'Task'
Object is of type 'Task'
Object is of type 'Cat2 ISR'
Object is of type 'Interval'
Object is of type 'Std resource'
Object is of type 'Linked resource'
...
```

シンタックス	開始 : <code>\$for:var=low,high\$</code> 終了 : <code>\$~for\$</code>
説明	<code>\$for:...\$</code> のループを処理する際、このプラグインはカプセル化されているトークンを、 <code>[low, high]</code> という範囲内の各整数につき 1 回解釈します。 カプセル化されているトークンが解釈されるたびに、変数 <code>var</code> にはループカウンタのカレント値が設定され、この変数は、レポート内において、通常の変数アクセスメカニズムによってアクセスできます。 <code>\$for:...\$</code> ループはネスティングが可能です。
例	以下のように記述すると、 <pre><code>\$for:z=1,3\$ "Line @z@ " \$for:x=1,@z@\$ "*" \$~for\$ \$nl\$ \$~for\$</code></pre> 以下のように出力されます。 <pre>Line 1 * Line 2 ** Line 3 ***</pre>

シンタックス	開始: <code>\$do:var=objects\$</code> 終了: <code>\$~do\$</code>
説明	<code>\$do:...\$</code> のループが処理される際、カプセル化されているトークンは、 <code>[low, high]</code> という範囲内の各整数につき 1 回解釈されます。 カプセル化されているトークンの解釈のたびに、マッチする各オブジェクト (マッチするオブジェクトが存在しない場合には "unknown object") のオブジェクトタイプと名前が変数 <code>var</code> に設定されます。 <code>\$do:...\$</code> ループはネスティング可能です。
例	以下のスクリプトは、タイムトレースビジュアライザに定義されているグループの内容を表示する処理を、すべてのグループについて繰り返し行います。 <pre>Groups:\$nl\$ \$repeat:i=groups\$ " @i@ members = @i.Members@" \$nl\$ \$do:m=@i.Members@\$ " @m@" \$nl\$ \$~do\$ \$nl\$ \$~repeat\$</pre> 典型的な出力は下のようになります。 <pre>Groups: Group 1 members = 9,3,7,5,16 Task tskLeader Task tskFollower Task tskInterfere Task tskIdle Cat2 ISR SystemISR</pre>

3.3.6 if

シンタックス	開始: <code>\$if:<lhs>,<cond>,<rhs>\$</code> 終了: <code>\$~if\$</code>
説明	このレポートマークアップ言語には、条件構文がサポートされています。ブロックにカプセル化されているトークンは、条件の評価結果が 'true' になった場合に限り解釈されます。 条件ブロックの中に他の条件構文をネストさせたり、 <code>\$repeat\$</code> や <code>\$for\$</code> のループの中に条件構文をネストさせたりすることもできます。 <cond> フィールドに設定できる値は <code>eq</code> 、 <code>ne</code> 、 <code>gt</code> 、 <code>lt</code> 、 <code>le</code> 、 <code>ge</code> です。
例	以下のように記述すると、システム内のオブジェクトについて、オブジェクトタイプを表示する処理が繰り返されます。オブジェクトの ID が 1000 より大きい場合には、(HTML タグを用いて) ボールド体で表示されます。 ... <code>\$repeat:ob=all\$</code> <code>\$if:@ob.ID@,gt,1000\$//start BOLD</code> <code>\$~if\$</code> @ob.Type@ <code>\$if:@ob.ID@,gt,1000\$// end BOLD</code> <code>\$~if\$</code> <code>\$nl\$</code> <code>\$~repeat\$</code> ...

3.3.7 nl

シンタックス	<code>\$nl\$</code>
説明	<code>\$nl\$</code> トークンはキャリッジリターン + ラインフィード (CRLF) のペアを出力ストリームに挿入します。これは、" <code>\n</code> " を使用する (つまり、キャリッジリターンだけを出力ストリームに挿入する) こととは異なります。

3.4 変数アクセス

このレポートマークアップ言語では、実行時に定義される変数へのアクセスを、`@variable@` というトークンにより行うことができます。

以下の変数は、トークンの解釈が始まる前に定義されます。

- `delim` – CSV レポートに使用されるデリミタです。これは Windows オペレーティングシステムから継承されるもので、「ロケール」に依存しています。ロケールのデシマルセパレータがカンマになっている場合には `delim` は「タブ」文字になり、そうでない場合には `delim` はカンマになります。
- `now` – システムのカレントロケールに従ってフォーマットされたカレント時刻です。
- `start` – “Multiple units” フォーマットを使用するレポート内の先頭レコードの時刻です。これがバッファ内の先頭レコードでもある場合には、時刻に“(start of trace data)”を付加します。
- `end` – “Multiple units” フォーマットを使用するレポート内の最終レコードの時刻です。これがバッファ内の最終レコードでもある場合には、時刻に“(end of trace data)”を付加します。
- `ns_per_tick` – 1 チックで表されるナノ秒数です。
- `product` – レポートジェネレータのロング名です。
- `version` – レポートプラグイン DLL の Windows のバージョン番号です。

3.5 オブジェクト属性と統計値の参照

オブジェクトは、トレース対象となるシステムの基本的なコンポーネント（タスク、ISR、リソースなど）です。オブジェクト属性を参照するには、`@Object.Attribute@` メカニズムが使用されます。

以下の形のオブジェクト参照が可能です（`tsk1` というタスクを例に説明します）。

- オブジェクトの名前（例：`tsk1`）
- オブジェクトの型と名前の組み合わせ（例：`Task tsk1`）
- `repeat` ループまたは `do` ループのループ変数¹ のための変数参照
- シノニム（3.6 項を参照してください。）

上記のように定義されたすべてのオブジェクト *Object* について、このプラグインは以下の属性の参照をサポートしています。

- **Name** – *Object* の名前を挿入します（たとえば、`@tsk1.Name@` のように指定すると出力ストリームに `tsk1` が出力されます）。
- **Type** – *Object* の型（"Task"、"Resource" など）を挿入します。
- **ID** – *Object* のオブジェクト識別子を挿入します。
- **ParentName** – *Object* のペアレントオブジェクトの名前を挿入します。*Object* のペアレントオブジェクトがない場合には `<none>` が挿入されます。
- **ParentType** – *Object* のペアレントオブジェクトの型を挿入します。このオブジェクトのペアレントオブジェクトがない場合には `<none>` が挿入されます。
- **ParentID** – *Object* のペアレントオブジェクトのオブジェクト識別子を挿入します。このオブジェクトのペアレントオブジェクトがない場合にはゼロ（"0"）が挿入されます。
- **Children** – *Object* に属するチャイルドオブジェクトの数を挿入します。
- **Members** – "groups" の場合に限り、グループのメンバーオブジェクトのオブジェクト識別子を1つずつカンマで区切ったリストを挿入します。

例 1 `tsk1` という名前のタスクがあるシステムで、以下のように記述すると

```
"Object '@tsk1.Name@' is of type
 '@obj.Type@'\n"
```

以下のように出力されます。

```
Object 'tsk1' is of type 'Task'
```

例 2 以下のように記述すると、

```
$repeat:obj=all$
"Object '@obj.Name@' is of type '@obj.Type@'\n"
$~repeat$
```

以下のように出力されます。

```
...
Object 'tsk1' is of type 'Task'
Object 'tsk2' is of type 'Task'
Object 'isr1' is of type 'Cat2 ISR'
Object 'int1' is of type 'Interval'
Object 'res1' is of type 'Std resource'
Object 'res2' is of type 'Linked resource'
...
```

また、`@Object.Statistic@` というトークンによる統計値の参照も行えます。

¹ これには `for` ループではなく `repeat` ループ用の変数参照が必要です。`for` ループのインデックスは整数なので、オブジェクト参照を確実に取得できるのは `repeat` ループだけであるためです。

テキスト **Statistic** は、統計値が定義されたときにユーザーがその統計値に与えた名前です。
テキスト **Statistic** には、'[と]' をデリミタとして使用してオブジェクト属性への参照を含めることができます。属性への参照は、統計値自体が参照される前に解決されます。以下に一例を紹介します。

```
$repeat:t=tasks$
  $repeat:r=std_resources$
    "@t.Name@ and resource @r.Name@: "
    $fmt:ms$"@t.Max Net Resource [r.Name]@ ms ("
    $fmt:str$"@t.Max Net Resource [r.Name]@"
    $nl$
  $~repeat$
$~repeat$
```

上記のように記述すると、システム内の各タスクのリソースごとの最大リソースロック時間が以下のように出力されます。

```
Task A and resource R1: 0.000400 ms (400ns)
Task B and resource R1: 0.000400 ms (400ns)
ISR Isr1 and resource R1: 0.000400 ms (400ns)
```

定義されていない属性または統計値を参照しようとすると、出力ファイルにエラーメッセージが出力されます。

3.6 オブジェクトのシノニム

システム内に一度は発生するけれども名前が前もって分からないオブジェクトについては、「シノニム」('synonym'= 同義語)を使用することによって標準アクセスメソッドを利用できるので、レポートテンプレート内でこれらのオブジェクトを容易に参照することができます。

レポートにおいては、システムオブジェクトをオブジェクト属性と統計値への参照に使用できるように、以下のシノニムを定義します。

- **System** – OS オブジェクト
- **Background** – アロケートされていないアクティビティ (アイドルタスク)
- **Error** – システムエラーメカニズム (フック)

4

お問い合わせ先

製品サポートに関しては、各 ETAS 支社までお問い合わせください。

ドイツ (ETAS 本社)

ETAS GmbH

Borsigstr. 14	Phone:	+49 (711) 8 96 61-0
70469 Stuttgart	Fax:	+49 (711) 8 96 61-105
Germany	E-mail:	sales@etas.de
	WWW:	www.etasgroup.com

日本

イータス株式会社

〒 220-6217	Phone:	(045) 222-0900
神奈川県横浜市西区	Fax:	(045) 222-0956
みなとみらい 2-3-5 クイーンズタワー C 17F	E-mail:	sales@etas.co.jp
	WWW:	www.etasgroup.com

韓国

ETAS Korea Co., Ltd.

3F Samseung Bldg	Phone:	+82 (2) 5747 016
61-1, Yangjae-dong	Fax:	+82 (2) 5747 120
Seocho-gu, Seoul	E-mail:	sungik.hong@etas.co.kr
Republic of Korea		

イギリス

ETAS Engineering Tools Application and Services Ltd.

Studio 3, Waterside Court	Phone:	+44 (0) 1283 - 546512
3rd Avenue, Centrum 100	Fax:	+44 (0) 1283 - 548767
Burton-upon-Trent	E-mail:	sales@etas-uk.net
Staffordshire DE14 2WQ	WWW:	www.etasgroup.com
UK		

フランス

ETAS S.A.S

1, place des Etats-Unis	Phone:	+33 (1) 56 70 00 50
SILIC 310	Fax:	+33 (1) 56 70 00 51
94588 Rungis Cedex	E-mail:	sales@etas.fr
France	WWW:	www.etasgroup.com

北米

ETAS Inc.

3021 Miller Road	Phone:	+1 (888) ETAS INC
Ann Arbor, MI 48103	Fax:	+1 (734) 997-9449
USA	E-mail:	sales@etasinc.com
	WWW:	www.etasgroup.com

南米

UNIT

Rua Adolfo Maraccini, 399
c/o Sergio Augusto Alves
Campinas SP 13086 - 010
Brazil

Mobile: +55 19 9772 0793
Telefax: +55 (19) 3256 1939
E-mail: unit@mpc.com.br

索引

D

do 16

F

fmt (format) 13

for 15

I

if 17

L

Looping

do 16

for 15

repeat 14

N

nl (newline) 17

O

Object 18

Children 18

ID 18

Name 18

ParentID 18

ParentName 18

ParentType 18
Type 18

R

repeat 14
 all 14
 group/groups 14
 interval/intervals 14
 isr/isrs 14
 process/processes 14
 profile/profiles 14
 resource/resources 14
 std_resource/std_resources 14
 task/tasks 14
 tracepoint/ tracepoints 14

S

Synonym 19
 Background 19
 Error 19
 System 19