

ALL-TIMES

ATF
All-times Trace Format



- ALL-TIMES -

ATF
All-times Trace Format

Version 1.0

License and Distribution:

The ATF schema and documentation is free software: you can redistribute it and/or modify it provided that modifications are submitted to atf-admin@all-times.org.

The logos used (if any) belong to their respective owners and should be used with prior written permission from the owner.



1. The ALL-TIMES project

ALL-TIMES is a medium-scale focused-research project within the European Commission's 7th Framework Programme on Research, Technological Development and Demonstration. It consists of six project partners from both industry and academia. Each project partner is an expert in its particular field and is also a provider of at least one tool for *timing analysis*.

More details to the ALL-TIMES project in general can be found on the website www.all-times.org.



2. Introduction

The ALL-TIMES Trace Format (ATF) is an XML file format that has been developed within the ALL-TIMES project, see <http://www.all-times.org>. Its purpose is to allow different software timing tools to exchange system configuration data, timing traces and timing measurement results. The system configuration describes all elements within a system that may consume CPU time, like tasks or runnables. A timing trace is a sequence of timestamped events that indicates how a system (in this case a software application) behaves over time. It is also possible to append measured timing results for each element in the system. So ATF also acts like a container for timing information that can be provided by any tool.

For one example, suppose that one event corresponds to the start of one task and another event corresponds to the start of another task and the tasks are supposed to run in strict alternation. A timing defect could be manifested as one of the tasks running twice in succession without the other task running in between. Even though this might be very rare and virtually impossible to capture using conventional debugging techniques, collecting and then searching a timing trace can highlight exactly when the problem occurs.

For another example, suppose that events correspond to the start and end of all the tasks and interrupt service routines (ISRs) in the application. A timing trace enables measurement of the execution time for each task and ISR individually, separating time spent actually executing a task from time spent while that task is not executing but waiting for an interrupt to complete.

Timing traces are a vital part of modern software development, debugging and verification. By having an open, publicly documented exchange format for traces, the ALL-TIMES partner tools allow sharing of trace data not only with each other but also with any other tools that can import or export ATF.

The following table lists the ALL-TIMES partners using ATF, the names of their tools and whether each tool imports or exports ATF.

<i>Partner</i>	<i>Tool</i>	<i>Import</i>	<i>Export</i>
Symtavision	SymTA/S	Viewer	
GLIWA	debugGURU	Viewer	OS and user defined events
Rapita Systems	RapiTime	Report constructor	OS and function events

The remainder of this document describes the hierarchical XML structure of ATF, starting with an overview, then progressing to a detailed field-by-field description and finishing with some illustrative examples. This documented is intended for ALL-TIMES project partners and any other tool builders who are constructing, or intend to construct, ATF interfaces.

If you wish to constructing a new ATF interface, please contact atf-admin@all-times.org for more information.



3. Schema overview

The general partition of the *ATF* is displayed in figure 1. The three major parts are:

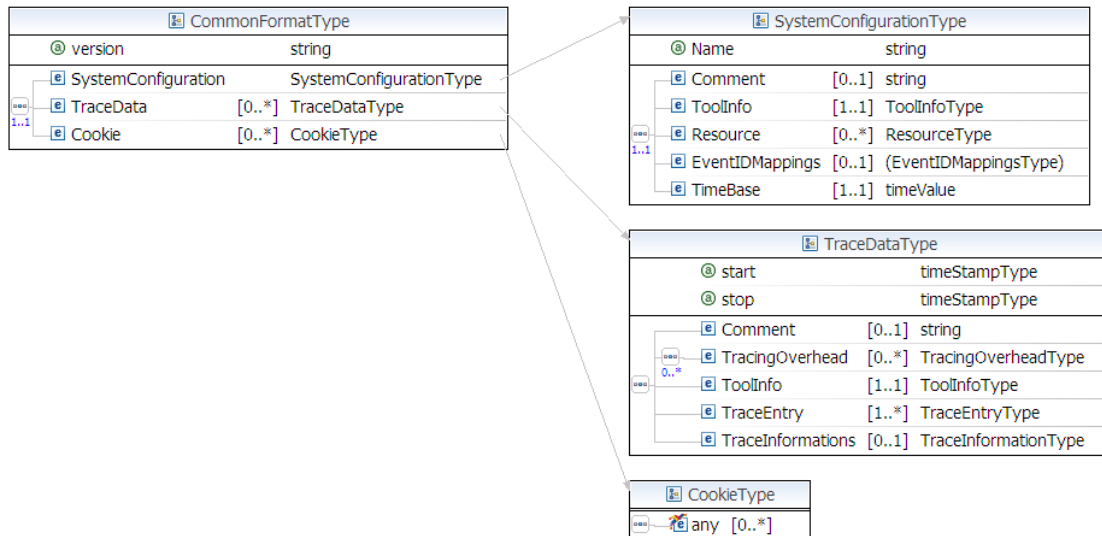


Figure 1: ATF Schema Overview

- *SystemConfiguration* (see page 16)
- *TraceData* (see page 26)
- *Cookie* (see page 8)

The *SystemConfiguration* can only be provided once per ATF. Here details to the system traced are provided (see page 14). Additionally configuration details to the trace events have to be provided (see page 9). So basically the tracing environment is configured here.

The *TraceData* can be provided multiple times. Each entry represents a separately taken trace. This is mainly the place for the pure trace information as provided in the TraceEntry (see page 27). Some additional information on the tracing overhead can be provided here as well (see page 30). Basically this is just the pure trace data.

The *Cookie* can also be provided multiple times and provides a possibility for tools to store additional internal data. Each tool processing an ATF file should leave the Cookie section of other tools in tact (or even copy them to new files if the trace data is copied). For sedatils see page 8.



4. Normative Description of ATF

Detailed descriptions for all types in the XML schema is provided in the subsection of this chapter. For an overview please see page 4.

The rest of this section list elements and the complex attribute types in alphabetical order. The root element, and a logical starting place for reading this section, is *Common-Format*, see page 7.



4.1. Annotation

The *Annotation* is a generic possibility to add user information to a traced element. This information is completely without semantic. Annotations provide a *name* and a *value*. These can be displayed as additional element information to the user by any tool.

There are some standardized *Annotation* elements. These are described in the appendix see *StandardizedAnnotations* on page 44.

4.1.1. Parent Elements

Annotation is a child of the *SystemElement* element, see page 18.

4.1.2. Attributes

Annotation has no Attributes.

4.1.3. Child Elements

Annotation has three child elements:

- *ToolInfo*, an optional element that represents the tool which made this annotation entry, see *ToolInfo* on page 25.
- *Name*, a string representing the name of some attribute value annotated to the *SystemElement*
- *Value*, a string

Where the same *name* string will most likely be used for many *SystemElement* (e.g. 'Priority' the values will be individual to each *SystemElement*).

4.1.4. Example

Example 1:

```
<Annotation>
  <Name>Priority</Name>
  <Value>7</Value>
</Annotation>
```

Example 2:

```
<Annotation>
  <ToolInfo Vendor="company X" Tool="tool Y" Version="x.y.z" />
  <Name>CETmax</Name>
  <Value>7000</Value>
</Annotation>
```



4.2. CommonFormat

The *CommonFormat* is the root element for the trace format.

4.2.1. Parent Elements

CommonFormat is the root element and does not have any parents.

4.2.2. Attributes

The only attribute for *CommonFormat* is *Version*. The attribute *Version* indicates the version of the ATF the file contains. To be conform with this document the tag '1.0' has to be used.

4.2.3. Child Elements

CommonFormat has tree child elements:

- *SystemConfiguration*, the system configuration that was traced, see page 16. Providing details on the traced elements.
- *TraceData*, each entry contains all traced events see *TraceData* on page 26
- *Cookie*, see *Cookie* on page 8

4.2.4. Example

```
<?xml version="1.0"?>
<CommonFormat xsi:noNamespaceSchemaLocation="AlltimesTraceFormat.xsd" Version="1.0">

  <SystemConfiguration Name="System1">
    </SystemConfiguration>

  <TraceData Start="0">
    </TraceData>

  <Cookie Vendor="Syntavision" Tool="TraceAnalyzer" Version="2.1">
    </Cookie>

</CommonFormat>
```



4.3. Cookie

The *Cookie* element allows any tool to cache additional information in the file. This is to ensure that a particular choice, for example about interpreting or processing the file, is retained and does not have to be repeated every time the file is accessed by that tool.

4.3.1. Parent Elements

Cookie is a child of the *CommonFormat* element, see page 7. It is derived from the *ToolInfo* element, see page 25.

4.3.2. Attributes

Cookie has exactly the same attributes as *ToolInfo* element, see page 25.

4.3.3. Child Elements

Cookie can have any child elements *no it can't*

4.3.4. Example

```
<Cookie Vendor="Rapita" Tool="RapiTime" Version="2.2" >  
  <RTD file="ex12.rtd" />  
</Cookie>
```




4.4. EventIDMapping

The *EventIDMapping* element maps from an event's *EventID* to the type of the event.

4.4.1. Parent Elements

EventIDMapping is a child of the *EventIDMappings* element, which is just a container for a list of a number of *EventIDMapping* elements. In turn, *EventIDMappings* is a child of the *SystemConfiguration* element, see page 16.

4.4.2. Attributes

EventIDMapping has two attributes:

- *EventID*, which is an unsigned integer. This is used in *TraceEntry* elements, see page 27.
- *EventType*, see page 10. It gives the kind of event, for example a *start* event.

4.4.3. Child Elements

If and only if *EventIDMapping* has *EventType* set to “user”, it has a *UserTable* child element, see *Info* on page 12.

4.4.4. Example

```
<EventIDMapping Time="1" EventID="1" />
```



4.5. EventTypeEnum

The type *EventTypeEnum* is an enumeration type.

4.5.1. Usage

EventTypeEnum is the type of the *EventType* attribute of element *EventIDMapping*, see page 9.

4.5.2. Values

activation-OS system element gets activated by the OS (periodically) or by an Interrupt

activation-chained system element gets activated by another system element

activation-failed failed activation of a system element

activation system element gets activated

error an error occurred and was traced as a error event

start system element starts running

stop system element stop working

preempt running system element gets preempted

resume preempted system element gets resumed

terminate system element gets terminated

user user event

4.5.3. Remarks

If we regard a trace event as recording “*what* happened *when* with *whom*”, the *EventType* attribute records the “*what*”.



4.6. Fraction

The *Fraction* is a convenient way to provide positive numbers that may contain decimals. A representation as Fraction allows calculations without losing accuracy. A major benefit is easy conversion from rate to speed.

4.6.1. Parent Elements

Fractions are used in *Time Value* (see page 24).

4.6.2. Attributes

There are two required attributes in a Fraction:

- *Numerator*, non negative integer
- *Denominator*, non negative integer

4.6.3. Children Elements

A Fraction does not contain any child element.

4.6.4. Example

```
<Fraction Numerator="1" Denominator="3"/>
```



4.7. Info

The *Info* element maps from a user event's *EventID* to the type of the user event.

4.7.1. Parent Elements

Info is a child of the *UserTable* element, which is just a container for a list of a number of *Info* elements. In turn, *UserTable* is a child of the *EventIDMapping* element, see page 9.

4.7.2. Attributes

Info has one attribute:

- *ReferenceID*, which is an unsigned integer. This is referenced from *TraceEntry* elements, see page 27.

4.7.3. Child Elements

Info has no child elements but it does have free text content.

4.7.4. Example

```
<Info ReferenceID="1">SYNC</Info>
```



4.8. OverheadType

The *OverheadType* is an enumeration type for the *OverheadType* attribute of element *TracingOverhead*, see page 30. Legal values include the following:

- *total overhead*
- *before taking time*
- *after taking time*



4.9. Resource

The *Resource* element represents a single event in the trace.

4.9.1. Parent Elements

Resource is a child of the *SystemConfiguration* element, see page 16.

4.9.2. Attributes

Resource has two attributes, all of which are required:

- *ID*, which is an unsigned integer unique to a particular resource
- *Scheduler*, which defines the type of scheduler, see *SchedulerType* on page 15

4.9.3. Child Elements

Resource has two child elements:

- *SystemElement*, see page 18
- *TimeRangeElement*, see *TimeRangeType* on page 21

4.9.4. Example

```
<Resource ID="33" Scheduler="OSEK">  
  <SystemElement Name="Task1" ID="99901" Type="task" />  
</Resource>
```



4.10. SchedulerType

The type *SchedulerType* is a string type.

4.10.1. Usage

The *SchedulerType* is the type of the *Scheduler* attribute of element *Resource*, see *Resource* on page 14.

4.10.2. Values

There are some predefined suggestions in the schema but any string value is possible.

Suggestions:

- CAN
- FlexRay
- LIN
- MOST
- AFDX
- ARINC 653
- OSEK
- ErcosEK
- AUTOSAR OS
- RTA OSEK
- GLI OS

4.10.3. Remarks

The *SchedulerType* is a string to hold the information about the scheduler of a *Resource* (see *Resource* on page 14).



4.11. SystemConfiguration

The *SystemConfiguration* describes all the elements the system contains. This information is required to resolve the data in the traces and the cookies.

4.11.1. Parent Elements

SystemConfiguration is a child of the *CommonFormat* element, see *CommonFormat* on page 7.

4.11.2. Attributes

It has an attribute *Name* with the name of the system.

4.11.3. Child Elements

The element *Comment* is optional and offers the possibility to add some comment text. The required element *ToolInfo* (see page 25) shows which tool has offered or created the system configuration.

A system can have 0..n resources (like different schedulers or subsystems) described in the listing *Resource* (see *Resource* on page 14).

To resolve the data of each trace entry the system configuration needs a mapping of event IDs to an event type. The element *EventIDMappings* contains this mapping table. The mapping itself is described by the element *EventIDMappingType* (see page 9). To resolve all timestamps in a trace the system configuration need a time base. This is described in the element *TimeBase* (see page 24).

4.11.4. Example

```
<SystemConfiguration Name="System1">
  <Comment>This is a very simple system configuration</Comment>
  <ToolInfo Vendor="Gliwa" Tool="T1" Version="1.5.45-2955" />
  <Resource ID="33" Scheduler="EDF">
    <SystemElement Name="Task1" ID="99901" Type="task" />
  </Resource>
  <EventIDMappings>
    <EventIDMapping EventID="1" EventType="start" />
    <EventIDMapping EventID="2" EventType="stop" />
  </EventIDMappings>
  <TimeBase Unit="s">
    <Value Numerator="1" Denominator="1" />
  </TimeBase>
</SystemConfiguration>
```




4.12. SystemElementType

The type *SystemElementType* is an enumeration type.

4.12.1. Usage

SystemElementType is the type of the *Type* attribute of element *SystemElement*, see page 18.

4.12.2. Values

isr interrupt service routine

task task

runnable runnable entity (AUTOSAR)

process global process (ERCOSEk) used in several tasks

function typically a C function but could also be an equivalent from another language, e.g. an Ada subprogram

basic block code that has one entry point, one exit point and no jump instructions anywhere else

message network or local message

unknown system element not expressible as any of the above

4.12.3. Remarks

The hierarchical nesting of *SystemElement* elements must conform to the restrictions shown in Figure 2. For example, a basic block can be within a function within a runnable within a task. However a message *SystemElement* cannot contain any other *SystemElement*. These constraints on nesting are not captured by the ATF schema and so cannot be enforced using schema validation.

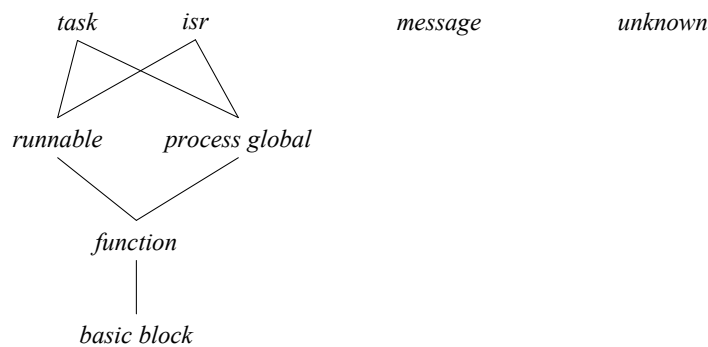


Figure 2: Valid nestings for hierachical *SystemElement* elements



4.13. SystemElement

The *SystemElement* represents some collection of executable code from which trace events arise.

4.13.1. Parent Elements

SystemElement is a child of the *Resource* element, see page 14.

4.13.2. Attributes

SystemElement has three attributes:

- *Name*, which is a string and is optional
- *ID*, which is an unsigned integer
- *SystemElement*, which defines the type of the system element, see page 18

4.13.3. Child Elements

SystemElement has three kinds of child element:

- *Comment*, zero or one string
- *Annotation*, any number, see *Annotation* on page 6
- *SystemElement*, any number, see *SystemElement* on page 18

Because *SystemElement* can be a child of another *SystemElement*, it allows a hierarchy of *SystemElements* to be defined, for example basic blocks within functions within tasks.

4.13.4. Example

```
<SystemElement Name="Task1" ID="99901" Type="task" />
```



4.14. TimeBaseType

The *TimeBaseType* is the unit component of the *TimeValue* (see page 24). The following units for are allowed: s, ms, us (for μs), ns, ps, as.



4.15. TimeRangeComment

The *TimeRangeComment* element is a comment linked to a particular range of times. It allows interesting events, or combination of events, in the trace to be annotated and for those annotations to be archived together with the trace data. The *TimeRangeComment* has no semantic information and is not interpreted by any of the tools. It is like a *TimeStampComment* but spans a range of time rather than just marking one moment in the trace, see page 22.

4.15.1. Parent Elements

TimeRangeComment is a child of the *TraceInformation* element, see page 28.

4.15.2. Attributes

TimeRangeComment has two attributes, all of which are required:

- *From*, which is a floating point number, see see *TimeStampType* on page 23. It is start time, from which the time range extends.
- *To*, which is a floating point number, see see *TimeStampType* on page 23. It is end time, to which the time range extends.
- *Text*, which is the comment itself.

4.15.3. Child Elements

TimeRangeComment has no child elements.

4.15.4. Example

```
<TimeRangeComment From="7" To="42" Text="look at this"/>
```



4.16. TimeRangeType

Declaration of different tracing custom types.

4.16.1. Usage

The *TimeRangeType* is type of the *Type* attribute of element *Resource*, see page 14.

4.16.2. Values

There are three possible values in this enumeration.

stopwatch Artificial element created for tracing and instrumentation of the target code.

worst case response time Element indicating the worst case response time traced.

data path This is not a single system element but rather a data element being handled by possibly multiple system elements. This allows for tracing data through a system.

4.16.3. Remarks

This can be used for multiple purposes during tracing. These elements allow to trace an event that does not belong to a *SystemElement* but can still be activated, preempted and so forth.



4.17. TimeStampComment

The *TimeStampComment* element is a comment linked to a particular timestamp. It allows interesting events, or combination of events, in the trace to be annotated and for those annotations to be archived together with the trace data. The *TimeStampComment* has no semantic information and is not interpreted by any of the tools. It is equivalent to a *TimeRangeComment* for which the *From* and *To* attributes are equal, see page 20.

4.17.1. Parent Elements

TimeStampComment is a child of the *TraceInformation* element, see page 28.

4.17.2. Attributes

TimeStampComment has two attributes, all of which are required:

- *At*, which is a floating point number, see *TimeStampType* on page 23. It is the time at which the comment applies.
- *Text*, which is the comment itself.

4.17.3. Child Elements

TimeStampComment has no child elements.

4.17.4. Example

```
<TimeStampComment At="50" Text="look at that" />
```



4.18. TimeStampType

The *TimeStampType* is to be clearly distinguished from *Fraction* (see page 11). The *TimeStampType* is used within the trace and does not contain a unit. The unit is later provided by use of the *TimeBase* in the *SystemConfiguration* (see page 16).

A *TimeStamp* is used to provide timing information in the trace. Normally this will be a “tick” count. Decimals are allowed to enable events between ticks or to provide support for non-tick based tools. The schema does not allow for negative or zero leading values.

4.18.1. Usage

The *TimeStamp* type is used in several places:

- *TimeStampComment*, see page 22.
- *TraceEntry*, see page 27.
- *TraceData*, see page 26.

4.18.2. Values

Positive numbers with decimals.

valid 0

1
42
3.141592

invalid 01

.5
1.2.3
-1

4.18.3. Remarks

Allowed strings for the values are guarded by an regular expression:

```
((0)|([1-9]([0-9])*))([\.]([0-9]+))?
```



4.19. TimeValue

The *TimeValue* is a definition of a specific point in time. This is not related to the *TimeBase* in the *SytemConfiguration* and therefore free of any “tick” used by the trace.

4.19.1. Parent Elements

The *TimeValue* is referenced by

- *TracingOverhead*, see page 30
- *SystemConfiguration*, see page 16

4.19.2. Attributes

The attribute '*Unit*' used by the the *TimeValue* is based on the *TimeBaseType*, see page 19.

4.19.3. Child Elements

The only allowed child element is *Fraction*, see page 11.

4.19.4. Example

```
<TimeValue Unit="ms">  
  <Value Numerator="1" Denominator="3"/>  
</TimeValue>
```




4.20. ToolInfo

The *ToolInfo* holds the information of the tool that provided the data (system configuration, trace or cookie).

4.20.1. Parent Elements

The *ToolInfo* element can have different parent elements. But all of these are parent elements can be seen as main elements. These are *SystemConfiguration* see *SystemConfiguration* on page 16, *TraceData* see *TraceData* on page 26 and *Cookie* see *Cookie* on page 8.

4.20.2. Attributes

The attributes to identify a tool are *Vendor*, *Tool* and *Version*.

4.20.3. Child Elements

ToolInfo has no child elements.

4.20.4. Example

```
<ToolInfo Vendor="company X" Tool="tool Y" Version="x.y" />
```



4.21. TraceData

The *TraceData* element is the main element of a trace. It contains all the trace information like all trace events, tool info and other trace information.

4.21.1. Parent Elements

TraceData is a child of the *CommonFormat* element, see *CommonFormat* on page 7.

4.21.2. Attributes

The element has two attributes *Start* and *Stop*, both of type *TimeStampType* see page 23. The *Start* attribute is required and defines the start time of the trace. The *Stop* attribute is optional and represents the end time.

4.21.3. Child Elements

The trace contains an optional element *Comment* to place some comment text. Like all main elements the trace requires an element *ToolInfo* (see *ToolInfo* on page 25) that shows which tool has created the trace data.

The most important data is the listing of all *TraceEntry* (see *TraceEntry* on page 27). This list contains all entries of a trace with the following information:

- When happened something?
- What happened?
- With whom?

To identify the measurement overhead of the whole tracing mechanism it is possible to define a list of overhead values in the element *TracingOverhead*. Each instrumentation routine has a different duration. So all these times can be defined in this list.

For additional specific comments the trace contains the element *TraceInformation* (see *TraceInformation* on page 28).

4.21.4. Example

```
<TraceData Start="0">
  <Comment>This is a very simple trace without tracing overhead or trace information</C
  <ToolInfo Vendor="Rapita" Tool="RapiTime" Version="2.2" />
  <TraceEntry Time="1" EventID="1" ReferenceID="99901" />
  <TraceEntry Time="2" EventID="1" ReferenceID="99901" />
  <TraceEntry Time="3" EventID="1" ReferenceID="99901" />
</TraceData>
```



4.22. TraceEntry

The *TraceEntry* element represents a single event in the trace.

4.22.1. Parent Elements

TraceEntry is a child of the *TraceData* element, see page 26.

4.22.2. Attributes

TraceEntry has three attributes, all of which are required:

- *Time*, which is a floating point number, see *TimeStampType* on page 23. It is the time at which the event occurred.
- *EventID*, which is an unsigned integer allocated in the *EventIDMapping* element, see page 9. It determines the kind of event that occurred, for example a *start* event.
- *ReferenceID*, which is an unsigned integer.
 - If the event is a user event, it indicates the kind of event and is allocated in the *UserTable* element, see *EventIDMapping* on page 9.
 - Otherwise, it indicates the *SystemElement*, see page 18, from which the *TraceEntry* arose.

4.22.3. Child Elements

TraceEntry has no child elements.

4.22.4. Example

```
<TraceEntry Time="1" EventID="1" ReferenceID="99901" />
```



4.23. TraceInformation

The *TraceInformation* is an area to add comments to the trace which can be used by tools to endorse the trace representation to the user.

4.23.1. Parent Elements

The comments belong to a trace and can therefore be added to a *TraceData* element. see page 26.

4.23.2. Attributes

There are no attributes for *TraceInformation*.

4.23.3. Children Elements

There are three types of comments available: *TimeRangeComment*, *TimeStampComment* and *Comment*. Please take a look at the type definitions:

TimeRangeComment comment for a range of time, see page 20

TimeStampComment comment for a point in time, see page 22

Comment general comment for the complete trace, any string

4.23.4. Example



4.24. TracingOverheadRangeType

The *TracingOverheadRangeType* is an enumeration type.

4.24.1. Usage

TracingOverheadRangeType is the type of the *Range* attribute of element *TracingOverhead*, see page 30.

4.24.2. Values

min min overhead value

max max overhead value

4.24.3. Remarks

The *TracingOverheadRangeType* is a simple enumeration with two possible value, see below. It's needed to specify if the accordant *TracingOverhead* is a min or max value (see *TracingOverhead* on page 30 for more information).



4.25. TracingOverhead

The *TracingOverhead* is an extension of the type *TimeValue* (see *timeValue* at 24). It defines a time value to identify and calculate the execution overhead the measurement or tracing causes. The value itself stands for the time that is needed to store one trace event during the measurement.

The type *TraceData* holds a list of tracing overhead values. So it is possible to specify each of the following options:

- option 1 (simplest):
1 time specifying the duration of one instrumentation event
- option 2: (still simple):
2 times for one time stamp:
– t1 –/– t2 –
At some point within the instrumentation routine, the system time is read. t1 specifies the time from the beginning of the routine up to that point, t2 from that point to the end.
- option 3:
foreach time t1 or t2 it is possible to specify a min and a max time. So we could have 4 times for one instrumentation routine.

4.25.1. Parent Elements

TracingOverhead is a part of the *TraceData* see *TraceData* on page 26.

4.25.2. Attributes

The extension of the *TimeValue* type are three optional attributes.

- *Overhead* (see *OverheadType* on page 13), defines if the given time belongs to option 1 or 2 (see below).
- *TracingOverheadRange* (see *TracingOverheadRangeType* on page 29), is needed if option 3 is chosen (see above).
- *ReferenceID*, which is an unsigned integer. It indicates the *SystemElement*, see page 18, for which the overhead is defined.

4.25.3. Child Elements

Because the *TracingOverhead* is an extension of the element *TimeValue* it has the same child element *Fraction* see *Fraction* on page 11.

4.25.4. Example

```
<TracingOverhead OverheadType="total overhead" Range="max" Unit="ms">
  <Value Numerator="1" Denominator="1000" />
</TracingOverhead>
```



5. Examples



5.1. Example 1

The following example shows a minimal system configuration with only one SystemElement (Task1), 2 event type (start and stop) and one trace with 4 events.

```
<?xml version="1.0"?>
<CommonFormat xsi:noNamespaceSchemaLocation="AlltimesTraceFormat.xsd"
    Version="0.2">
  <SystemConfiguration Name="System1">
    <ToolInfo Vendor="Rapita" Tool="RapiTime" Version="2.2" />
    <Resource ID="33" Scheduler="EDF">
      <SystemElement Name="Task1" ID="99901" Type="task" />
    </Resource>
    <EventIDMappings>
      <EventIDMapping EventID="1" EventType="start" />
      <EventIDMapping EventID="2" EventType="end" />
    </EventIDMappings>
    <TimeBase Unit="s">
      <Value Numerator="1" Denominator="1" />
    </TimeBase>
  </SystemConfiguration>
  <TraceData Start="0">
    <ToolInfo Vendor="Rapita" Tool="RapiTime" Version="2.2" />
    <TraceEntry Time="1" EventID="1" ReferenceID="99901" />
    <TraceEntry Time="2" EventID="2" ReferenceID="99901" />
    <TraceEntry Time="4" EventID="1" ReferenceID="99901" />
    <TraceEntry Time="5" EventID="2" ReferenceID="99901" />
  </TraceData>
</CommonFormat>
```




5.2. Example 2

The following example shows 2 start and stops of one task and some user events.

```
<?xml version="1.0" encoding="utf-8"?>
<CommonFormat xsi:noNamespaceSchemaLocation="AlltimesTraceFormat.xsd"
    Version="0.2">
  <SystemConfiguration Name="System1">
    <ToolInfo Vendor="Gliwa GmbH" Tool="T1" Version="1.5.18-beta2293" />
    <Resource Scheduler="OSEK" ID="0">
      <SystemElement Name="Task1" ID="99901" Type="task">
        <Comment>This is Task1</Comment>
        <Annotation>
          <Name>Priority</Name>
          <Value>0</Value>
        </Annotation>
      </SystemElement>
    </Resource>
    <EventIDMappings>
      <EventIDMapping EventID="1" EventType="start" />
      <EventIDMapping EventID="3" EventType="end" />
      <EventIDMapping EventID="4" EventType="end" />
      <EventIDMapping EventID="5" EventType="resume" />
      <EventIDMapping EventID="6" EventType="preempt" />
      <EventIDMapping EventID="11" EventType="error" />
      <EventIDMapping EventID="2" EventType="user">
        <UserTable>
          <Info ReferenceID="1">SYNC</Info>
          <Info ReferenceID="2">Info 1</Info>
        </UserTable>
      </EventIDMapping>
    </EventIDMappings>
    <TimeBase Unit="ns">
      <Value Numerator="500000000" Denominator="1" />
    </TimeBase>
  </SystemConfiguration>
  <TraceData Start="0">
    <Comment>tracel</Comment>
    <ToolInfo Vendor="Gliwa GmbH" Tool="T1" Version="1.5.18-beta2293" />
    <TraceEntry Time="2" EventID="2" ReferenceID="1" />
    <TraceEntry Time="3" EventID="2" ReferenceID="2" />
    <TraceEntry Time="4" EventID="1" ReferenceID="99901" />
    <TraceEntry Time="10" EventID="3" ReferenceID="99901" />
    <TraceEntry Time="12" EventID="2" ReferenceID="3" />
    <TraceEntry Time="22" EventID="2" ReferenceID="1" />
    <TraceEntry Time="23" EventID="2" ReferenceID="2" />
    <TraceEntry Time="24" EventID="1" ReferenceID="99901" />
    <TraceEntry Time="30" EventID="3" ReferenceID="99901" />
    <TraceEntry Time="32" EventID="2" ReferenceID="3" />
  </TraceData>
</CommonFormat>
```



</TraceData>
</CommonFormat>



5.3. Example 3

The following example shows a preemption/resumption of a task.

```
<?xml version="1.0" encoding="utf-8"?>
<CommonFormat xsi:noNamespaceSchemaLocation="AlltimesTraceFormat.xsd"
    Version="0.2">
  <SystemConfiguration Name="System1">
    <ToolInfo Vendor="Gliwa GmbH" Tool="T1" Version="1.5.18-beta2293" />
    <Resource Scheduler="OSEK" ID="0">
      <SystemElement Name="Task1" ID="99901" Type="task">
        <Comment>This is Task1</Comment>
        <Annotation>
          <Name>Priority</Name>
          <Value>0</Value>
        </Annotation>
      </SystemElement>
      <SystemElement Name="Task2" ID="99902" Type="task">
        <Comment>This is Task2</Comment>
        <Annotation>
          <Name>Priority</Name>
          <Value>2</Value>
        </Annotation>
      </SystemElement>
      <SystemElement Name="Task3" ID="99903" Type="task">
        <Comment>This is Task3</Comment>
        <Annotation>
          <Name>Priority</Name>
          <Value>3</Value>
        </Annotation>
      </SystemElement>
    </Resource>
    <EventIDMappings>
      <EventIDMapping EventID="1" EventType="start" />
      <EventIDMapping EventID="3" EventType="end" />
      <EventIDMapping EventID="5" EventType="resume" />
      <EventIDMapping EventID="6" EventType="preempt" />
      <EventIDMapping EventID="11" EventType="error" />
      <EventIDMapping EventID="2" EventType="user">
        <UserTable>
          <Info ReferenceID="1">SYNC</Info>
          <Info ReferenceID="2">Info 1</Info>
        </UserTable>
      </EventIDMapping>
    </EventIDMappings>
    <TimeBase Unit="ns">
      <Value Numerator="500000000" Denominator="1" />
    </TimeBase>
  </SystemConfiguration>
```



```
<TraceData Start="0">
  <Comment>trace1</Comment>
  <ToolInfo Vendor="Gliwa GmbH" Tool="T1" Version="1.5.18-beta2293" />
  <TraceEntry Time="2" EventID="2" ReferenceID="1" />
  <TraceEntry Time="3" EventID="2" ReferenceID="2" />
  <TraceEntry Time="4" EventID="1" ReferenceID="99901" />
  <TraceEntry Time="6" EventID="6" ReferenceID="99901" />
  <TraceEntry Time="6" EventID="1" ReferenceID="99902" />
  <TraceEntry Time="7" EventID="3" ReferenceID="99902" />
  <TraceEntry Time="7" EventID="5" ReferenceID="99901" />
  <TraceEntry Time="10" EventID="3" ReferenceID="99901" />
  <TraceEntry Time="12" EventID="2" ReferenceID="3" />
  <TraceEntry Time="22" EventID="2" ReferenceID="1" />
  <TraceEntry Time="23" EventID="2" ReferenceID="2" />
  <TraceEntry Time="24" EventID="1" ReferenceID="99901" />
  <TraceEntry Time="30" EventID="3" ReferenceID="99901" />
  <TraceEntry Time="32" EventID="2" ReferenceID="3" />
</TraceData>
</CommonFormat>
```



5.4. Example 4

The following example shows that it's possible to have multiple traces in one file.

```
<?xml version="1.0" encoding="utf-8"?>
<CommonFormat xsi:noNamespaceSchemaLocation="AlltimesTraceFormat.xsd"
    Version="0.2">
  <SystemConfiguration Name="System1">
    <ToolInfo Vendor="Gliwa GmbH" Tool="T1" Version="1.5.18-beta2293" />
    <Resource Scheduler="OSEK" ID="0">
      <SystemElement Name="Task1" ID="99901" Type="task">
        <Comment>This is Task1</Comment>
        <Annotation>
          <Name>Priority</Name>
          <Value>0</Value>
        </Annotation>
      </SystemElement>
      <SystemElement Name="Task2" ID="99902" Type="task">
        <Comment>This is Task2</Comment>
        <Annotation>
          <Name>Priority</Name>
          <Value>2</Value>
        </Annotation>
      </SystemElement>
      <SystemElement Name="Task3" ID="99903" Type="task">
        <Comment>This is Task3</Comment>
        <Annotation>
          <Name>Priority</Name>
          <Value>3</Value>
        </Annotation>
      </SystemElement>
    </Resource>
    <EventIDMappings>
      <EventIDMapping EventID="1" EventType="start" />
      <EventIDMapping EventID="3" EventType="end" />
      <EventIDMapping EventID="5" EventType="resume" />
      <EventIDMapping EventID="6" EventType="preempt" />
      <EventIDMapping EventID="11" EventType="error" />
      <EventIDMapping EventID="2" EventType="user">
        <UserTable>
          <Info ReferenceID="1">SYNC</Info>
          <Info ReferenceID="2">Info 1</Info>
        </UserTable>
      </EventIDMapping>
    </EventIDMappings>
    <TimeBase Unit="ns">
      <Value Numerator="500000000" Denominator="1" />
    </TimeBase>
  </SystemConfiguration>
```



```
<TraceData Start="0">
  <Comment>trace1</Comment>
  <ToolInfo Vendor="Gliwa GmbH" Tool="T1" Version="1.5.18-beta2293" />
  <TraceEntry Time="2" EventID="2" ReferenceID="1" />
  <TraceEntry Time="3" EventID="2" ReferenceID="2" />
  <TraceEntry Time="4" EventID="1" ReferenceID="99901" />
  <TraceEntry Time="6" EventID="6" ReferenceID="99901" />
  <TraceEntry Time="6" EventID="1" ReferenceID="99902" />
  <TraceEntry Time="7" EventID="3" ReferenceID="99902" />
  <TraceEntry Time="7" EventID="5" ReferenceID="99901" />
  <TraceEntry Time="10" EventID="3" ReferenceID="99901" />
  <TraceEntry Time="12" EventID="2" ReferenceID="3" />
  <TraceEntry Time="22" EventID="2" ReferenceID="1" />
  <TraceEntry Time="23" EventID="2" ReferenceID="2" />
  <TraceEntry Time="24" EventID="1" ReferenceID="99901" />
  <TraceEntry Time="30" EventID="3" ReferenceID="99901" />
  <TraceEntry Time="32" EventID="2" ReferenceID="3" />
</TraceData>
<TraceData Start="0">
  <Comment>trace2</Comment>
  <ToolInfo Vendor="Gliwa GmbH" Tool="T1" Version="1.5.18-beta2293" />
  <TraceEntry Time="2" EventID="2" ReferenceID="1" />
  <TraceEntry Time="3" EventID="2" ReferenceID="2" />
  <TraceEntry Time="4" EventID="1" ReferenceID="99901" />
  <TraceEntry Time="10" EventID="3" ReferenceID="99901" />
  <TraceEntry Time="12" EventID="2" ReferenceID="3" />
  <TraceEntry Time="22" EventID="2" ReferenceID="1" />
  <TraceEntry Time="23" EventID="2" ReferenceID="2" />
  <TraceEntry Time="24" EventID="1" ReferenceID="99901" />
  <TraceEntry Time="26" EventID="6" ReferenceID="99901" />
  <TraceEntry Time="26" EventID="1" ReferenceID="99902" />
  <TraceEntry Time="27" EventID="3" ReferenceID="99902" />
  <TraceEntry Time="27" EventID="5" ReferenceID="99901" />
  <TraceEntry Time="30" EventID="3" ReferenceID="99901" />
  <TraceEntry Time="32" EventID="2" ReferenceID="3" />
</TraceData>
</CommonFormat>
```



5.5. Example 5

This is a little bit more complex trace with some tasks and user events. There's also an measurement overhead defined in the trace.

```
<?xml version="1.0" encoding="utf-8"?>
<CommonFormat xsi:noNamespaceSchemaLocation="AlltimesTraceFormat.xsd"
    Version="0.2">
  <SystemConfiguration Name="yahobnode">
    <Comment>no comment</Comment>
    <ToolInfo Vendor="Gliwa GmbH" Tool="T1" Version="1.5.18-beta2293" />
    <Resource Scheduler="GLI OS" ID="0">
      <SystemElement Name="OS_ISR" ID="221" Type="isr" />
      <SystemElement Name="my10msTask" ID="4" Type="task" />
      <SystemElement Name="my100msTask" ID="8" Type="task" />
      <SystemElement Name="debugGuruTask" ID="16" Type="task" />
      <SystemElement Name="ledTask" ID="1" Type="task" />
      <SystemElement Name="init" ID="126" Type="task" />
      <SystemElement Name="backGround" ID="127" Type="task" />
    </Resource>
    <EventIDMappings>
      <EventIDMapping EventID="1" EventType="activation" />
      <EventIDMapping EventID="2" EventType="activation-failed" />
      <EventIDMapping EventID="3" EventType="start" />
      <EventIDMapping EventID="4" EventType="end" />
      <EventIDMapping EventID="5" EventType="resume" />
      <EventIDMapping EventID="6" EventType="preempt" />
      <EventIDMapping EventID="11" EventType="error" />
      <EventIDMapping EventID="0" EventType="user">
        <UserTable />
      </EventIDMapping>
      <EventIDMapping EventID="9" EventType="user">
        <UserTable />
      </EventIDMapping>
      <EventIDMapping EventID="12" EventType="user">
        <UserTable />
      </EventIDMapping>
      <EventIDMapping EventID="15" EventType="user">
        <UserTable />
      </EventIDMapping>
      <EventIDMapping EventID="16" EventType="user">
        <UserTable />
      </EventIDMapping>
      <EventIDMapping EventID="18" EventType="user">
        <UserTable />
      </EventIDMapping>
      <EventIDMapping EventID="195" EventType="user">
        <UserTable />
      </EventIDMapping>
    </EventIDMappings>
  </SystemConfiguration>
</CommonFormat>
```



```
</EventIDMappings>
<TimeBase Unit="ns">
  <Value Numerator="2000" Denominator="1" />
</TimeBase>
</SystemConfiguration>
<TraceData Start="0" Stop="23038">
  <Comment>trace1</Comment>
  <TracingOverhead OverheadType="total overhead" Unit="ns">
    <Value Numerator="400" Denominator="1" />
  </TracingOverhead>
  <ToolInfo Vendor="Gliwa GmbH" Tool="T1" Version="1.5.18-beta2293" />
  <TraceEntry Time="0" EventID="3" ReferenceID="16" />
  <TraceEntry Time="48" EventID="4" ReferenceID="16" />
  <TraceEntry Time="105" EventID="195" ReferenceID="1" />
  <TraceEntry Time="883" EventID="195" ReferenceID="1" />
  <TraceEntry Time="903" EventID="1" ReferenceID="4" />
  <TraceEntry Time="999" EventID="3" ReferenceID="4" />
  <TraceEntry Time="1017" EventID="4" ReferenceID="4" />
  <TraceEntry Time="1074" EventID="195" ReferenceID="1" />
  <TraceEntry Time="2377" EventID="195" ReferenceID="1" />
  <TraceEntry Time="2401" EventID="1" ReferenceID="16" />
  <TraceEntry Time="2495" EventID="3" ReferenceID="16" />
  <TraceEntry Time="2545" EventID="4" ReferenceID="16" />
  <TraceEntry Time="2602" EventID="195" ReferenceID="1" />
  <TraceEntry Time="3128" EventID="195" ReferenceID="0" />
  <TraceEntry Time="3313" EventID="195" ReferenceID="4" />
  <TraceEntry Time="3480" EventID="195" ReferenceID="3" />
  <TraceEntry Time="3741" EventID="195" ReferenceID="5" />
  <TraceEntry Time="4900" EventID="1" ReferenceID="16" />
  <TraceEntry Time="4993" EventID="3" ReferenceID="16" />
  <TraceEntry Time="5044" EventID="4" ReferenceID="16" />
  <TraceEntry Time="5101" EventID="195" ReferenceID="1" />
  <TraceEntry Time="5227" EventID="195" ReferenceID="1" />
  <TraceEntry Time="5879" EventID="195" ReferenceID="1" />
  <TraceEntry Time="5900" EventID="1" ReferenceID="4" />
  <TraceEntry Time="5995" EventID="3" ReferenceID="4" />
  <TraceEntry Time="6013" EventID="4" ReferenceID="4" />
  <TraceEntry Time="6070" EventID="195" ReferenceID="1" />
  <TraceEntry Time="6196" EventID="195" ReferenceID="1" />
  <TraceEntry Time="7398" EventID="1" ReferenceID="16" />
  <TraceEntry Time="7491" EventID="3" ReferenceID="16" />
  <TraceEntry Time="7541" EventID="4" ReferenceID="16" />
  <TraceEntry Time="7598" EventID="195" ReferenceID="0" />
  <TraceEntry Time="7784" EventID="195" ReferenceID="4" />
  <TraceEntry Time="7953" EventID="195" ReferenceID="3" />
</TraceData>
</CommonFormat>
```




5.6. Example 6

This example shows a simple hierarchical SystemElements structure. The *SystemElement debugGuruTask* has two child elements *debugGURUProcess_startHandler* and *debugGURUProcess_endHandler*.

```
<?xml version="1.0" encoding="utf-8"?>
<CommonFormat xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xsi:noNamespaceSchemaLocation="AlltimesTraceFormat.xsd"
  Version="0.2">
  <SystemConfiguration Name="yahobnode">
    <Comment>no comment</Comment>
    <ToolInfo Vendor="Gliwa GmbH" Tool="T1" Version="1.5.18-beta2293" />
    <Resource Scheduler="GLI OS" ID="0">
      <SystemElement Name="OS_ISR" ID="221" Type="isr">
        <Annotation>
          <Name>Priority</Name>
          <Value>100</Value>
        </Annotation>
      </SystemElement>
      <SystemElement Name="my10msTask" ID="4" Type="task">
        <Annotation>
          <Name>Priority</Name>
          <Value>3</Value>
        </Annotation>
      </SystemElement>
      <SystemElement Name="my100msTask" ID="8" Type="task">
        <Annotation>
          <Name>Priority</Name>
          <Value>2</Value>
        </Annotation>
      </SystemElement>
      <SystemElement Name="debugGuruTask" ID="16" Type="task">
        <Annotation>
          <Name>Priority</Name>
          <Value>1</Value>
        </Annotation>
        <SystemElement Name="debugGURUProcess_startHandler" ID="256" Type="runnable" />
        <SystemElement Name="debugGURUProcess_endHandler" ID="257" Type="runnable" />
      </SystemElement>
      <SystemElement Name="ledTask" ID="1" Type="task">
        <Annotation>
          <Name>Priority</Name>
          <Value>5</Value>
        </Annotation>
      </SystemElement>
      <SystemElement Name="init" ID="126" Type="task">
        <Annotation>
```



```

        <Name>Priority</Name>
        <Value>10</Value>
    </Annotation>
</SystemElement>
    <SystemElement Name="backGround" ID="127" Type="task">
        <Annotation>
            <Name>Priority</Name>
            <Value>0</Value>
        </Annotation>
    </SystemElement>
</Resource>
<EventIDMappings>
    <EventIDMapping EventID="1" EventType="activation" />
    <EventIDMapping EventID="2" EventType="activation-failed" />
    <EventIDMapping EventID="3" EventType="start" />
    <EventIDMapping EventID="4" EventType="end" />
    <EventIDMapping EventID="5" EventType="resume" />
    <EventIDMapping EventID="6" EventType="preempt" />
    <EventIDMapping EventID="11" EventType="error" />
</EventIDMappings>
<TimeBase Unit="ns">
    <Value Numerator="2000" Denominator="1" />
</TimeBase>
</SystemConfiguration>
<TraceData Start="0" Stop="23038">
    <Comment>trace1</Comment>
    <TracingOverhead OverheadType="total overhead" Unit="ns">
        <Value Numerator="400" Denominator="1" />
    </TracingOverhead>
    <ToolInfo Vendor="Gliwa GmbH" Tool="T1" Version="1.5.18-beta2293" />
    <TraceEntry Time="0" EventID="3" ReferenceID="16" />
    <TraceEntry Time="48" EventID="4" ReferenceID="16" />
    <TraceEntry Time="903" EventID="1" ReferenceID="4" />
    <TraceEntry Time="999" EventID="3" ReferenceID="4" />
    <TraceEntry Time="1017" EventID="4" ReferenceID="4" />
    <TraceEntry Time="2401" EventID="1" ReferenceID="16" />
    <TraceEntry Time="2495" EventID="3" ReferenceID="16" />
    <TraceEntry Time="2529" EventID="6" ReferenceID="16" />
    <TraceEntry Time="2529" EventID="3" ReferenceID="221" />
    <TraceEntry Time="2590" EventID="4" ReferenceID="221" />
    <TraceEntry Time="2590" EventID="5" ReferenceID="16" />
    <TraceEntry Time="2601" EventID="3" ReferenceID="256" />
    <TraceEntry Time="2650" EventID="4" ReferenceID="256" />
    <TraceEntry Time="2750" EventID="3" ReferenceID="257" />
    <TraceEntry Time="2789" EventID="4" ReferenceID="257" />
    <TraceEntry Time="2945" EventID="4" ReferenceID="16" />
    <TraceEntry Time="4900" EventID="1" ReferenceID="16" />

```



```
<TraceEntry Time="4993" EventID="3" ReferenceID="16" />
<TraceEntry Time="4999" EventID="3" ReferenceID="256" />
<TraceEntry Time="5010" EventID="4" ReferenceID="256" />
<TraceEntry Time="5011" EventID="3" ReferenceID="257" />
<TraceEntry Time="5040" EventID="4" ReferenceID="257" />
<TraceEntry Time="5044" EventID="4" ReferenceID="16" />
<TraceEntry Time="5900" EventID="1" ReferenceID="4" />
<TraceEntry Time="5995" EventID="3" ReferenceID="4" />
<TraceEntry Time="6013" EventID="4" ReferenceID="4" />
<TraceEntry Time="7398" EventID="1" ReferenceID="16" />
<TraceEntry Time="7491" EventID="3" ReferenceID="16" />
<TraceEntry Time="7499" EventID="3" ReferenceID="256" />
<TraceEntry Time="7510" EventID="4" ReferenceID="256" />
<TraceEntry Time="7511" EventID="3" ReferenceID="257" />
<TraceEntry Time="7540" EventID="4" ReferenceID="257" />
<TraceEntry Time="7541" EventID="4" ReferenceID="16" />
</TraceData>
</CommonFormat>
```



A. Standardized Annotation Elements

In some cases tools need additional information of a *SystemElement* like the priority or timing values. So there are a couple of *Annotation* elements which these tools know and which they can interpret.

Values are:

- *Priority* holds the priority information of a *SystemElement*

```
<Annotation>
  <Name>Priority</Name>
  <Value>7</Value>
</Annotation>
```

- *CET* core execution time (see *TimingInformationValues* on page 46)
 - *CETmax* holds the maximum core execution time of a *SystemElement*. The time base of the *Value* is defined in the element *TimeBase* of *SystemConfiguration*, see *SystemConfiguration* on page 16.

```
<Annotation>
  <Name>CETmax</Name>
  <Value>720</Value>
</Annotation>
```

- *CETmin* holds the minimum core execution time of a *SystemElement* (time base see *CETmax*).

```
<Annotation>
  <Name>CETmin</Name>
  <Value>450</Value>
</Annotation>
```

- *CETav* holds the average core execution time of a *SystemElement* (time base see *CETmax*).

```
<Annotation>
  <Name>CETav</Name>
  <Value>660</Value>
</Annotation>
```

- *GET* gross execution time (see *TimingInformationValues* on page 46)

- *GETmax*
- *GETmin*
- *GETav*

- *IPT* initial pending time (see *TimingInformationValues* on page 46)

- *IPTmax*
- *IPTmin*
- *IPTav*

- *RT* response time (see *TimingInformationValues* on page 46)



- RT_{max}
- RT_{min}
- RT_{av}
- DT delta time (see *TimingInformationValues* on page 46)
 - DT_{max}
 - DT_{min}
 - DT_{av}
- PER period (see *TimingInformationValues* on page 46)
 - PER_{max}
 - PER_{min}
 - PER_{av}
- ST slack time (see *TimingInformationValues* on page 46)
 - ST_{max}
 - ST_{min}
 - ST_{av}
- JIT jitter (see *TimingInformationValues* on page 46)
 - JIT_{max}
 - JIT_{min}
 - JIT_{av}
- PRE preemption (see *TimingInformationValues* on page 46)
 - PRE_{max}
 - PRE_{min}
 - PRE_{av}



B. Timing Information Values

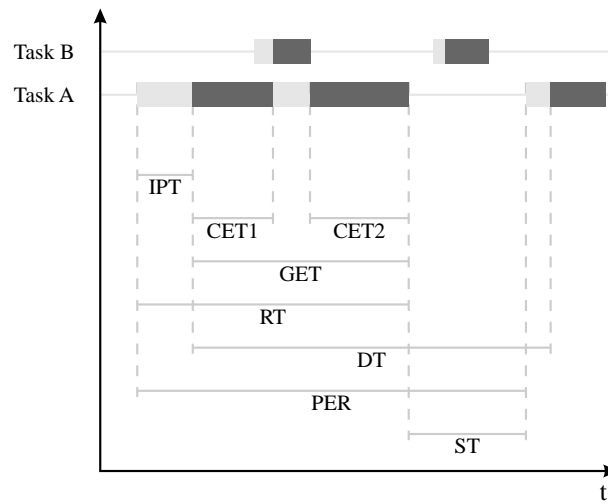


Figure 3: Timing results visualised in a trace (all related to TASK A)



ID	Abr.	Name EN	Name DE	Description
1	IPT	initial pending time	Initialwartezeit	from activation to start
2	CET	core execution time	Nettolaufzeit	execution time not including any preemptions
3	GET	gross execution time	Bruttolaufzeit	execution time including all preemptions
4	RT	response time	Antwortzeit	from activation to termination
5	DT	delta time	Deltazeit	from start to start ("measured period")
6	PER	period	Periode	from activation to activation (period not as measured but as configured)
7	ST	slack time	Restzeit	"remaining" run-time: from termination to activation (tasks) or start (interrupts)
8	JIT	jitter $\left(1 - \frac{DT}{PER}\right)$	Jitter	deviation of delta time from period
9	PRE	preemption	Unterbrechung	from preemption to resumption

Table 1: Timing results