



Contribution ID: 137

Type: **not specified**

## Context of session - and proposed agenda

*Monday 19 October 2009 14:10 (10 minutes)*

**\*\* Focus \*\***

Proposed as a working session - focusing on:

- \* understanding the current use cases for Physics tables (in sequential Geant4 applications)
- \* understanding the characteristics (and limitations) when used in a multi-process (copy on write) application;
- \* understanding the characteristics and requirements in the use case of a prototype multi-threaded Geant4 application;
- \* clarifying the requirements for Physics table(s) coming from multi-process and multi-core extensions;
- \* discussing potential ways to overcome these current weaknesses
- \* summarising our considerations, as a starting point for a deeper examination and potential revision of the design of the Physics tables to cope with the new requirements.

### Summary

Potential agenda

\* Brief introduction to Physics Tables

( focusing on use cases, requirements ) - talk of 3-4 slides, ( ~ 5 minutes )

- Memory cost of Physics tables

\*Considerations and challenges of multi-core ( ~ 30 minutes )

Mini-presentation: Maximum 10 slides [ 15 minutes ]

- Why share physics tables - either as pages of memory or as objects ?

- Current difficulties in multi-core (mixing of read-only and read-write addresses)

- How are physics tables (and client classes) treated in Xin's Multi-Threaded prototype

Discussion [ ~ 15 minutes ]

- Challenge of multi-threading (explanation of context in M/T prototype)

- Are there potential benefits for ordinary G4 (sequential) from revising the PhysicsTable implementation?

\* Clarification of use case(s) - with focus on multi-core/multi-threading ( ~ 20 minutes )

\* Distillation of requirements, potential constraints ( ~ 20 minutes )

\* Potential solutions ( ~ 10 minutes )

**Session Classification:** Parallel Session I - Particle tables and processes