

Draft Java Specification Request

Title: Safety Critical Java Technology

Summary: This specification creates a J2ME building block, based on the RTSJ, containing the minimal features necessary for safety critical real-time systems capable of certification, e.g., DO-178B / ED-12B.

Submitting Member: ?

Name of Contact Person: ?

E-Mail Address: ?

Telephone Number: ?

Fax Number: ?

Name of Specification Lead: The Open Group (Joe Bergmann)

E-Mail Address: ?

Telephone Number: ?

Fax Number: ?

Initial Group Membership: ?

Supporting this JSR: ?

Description: The proposed specification will define those capabilities needed to create safety critical applications. This means that the features included will be a minimal set, with such specific characteristics as static resource allocation and usage, minimal temporal conflicts, and without dynamic loading, leading to the ability to validate implementations using a variety of standards, including DO-178B / ED-12B. It is further implied that the features chosen can be validated using formal models, schedulability analysis, and modified condition/decision coverage (MC/DC) analysis.

It is strongly intended that this specification will incorporate the existing Java paradigm maximally, subject to the need for application validation. Most safety critical standards require fully predetermined resource allocation, thus necessitating these requirements. For example, a garbage collector cannot be used under such standards, and components cannot be dynamically loaded. Such applications will likely require a transformation from Java bytecodes to target machine representation prior to certification.

The Specification Lead will produce a TCK that will test only the resulting specification.

All application programs that successfully execute on the Reference Implementation will also execute on any Java platform subject to availability of suitable libraries.

This JSR will result in a specification such that ~~the International Organization for~~ Standardization (ISO) could later accept it.

Deleted: will be independent of other specifications so that

Deleted: Standards

Target Platform: J2ME

As of 23 July 04

Java community need: Safety critical systems need a certifiable (e.g., DO-178B) Java environment. Certifiability implies hard real-time resource management and generally very small implementations with low complexity.

Why this need isn't met now: The existing RTSJ specification (JSR-000001) contains both too many functions, and functions that are more complex than can be made certifiable.

Underlying Java technologies: RTSJ (JSR-000001), J2ME

Proposed Package Name for the API Specification: [javax.scsj](#)

Deleted: Safety Critical Specification for Java (SCSJ)

Does proposed spec. have dependencies on specific OSs, CPUs, or devices? No.

Are there security issues not addressed by current security model? No.

Internationalization or Localization issues? No.

Existing specifications that might be rendered obsolete, deprecated, or in need of revision if this JSR is successful? If this Expert Group determines that the SCSJ differs from a proper subset of the RTSJ, the RTSJ should be revised to incorporate the changes in coordination with the proposed JSR's Expert Group.

Schedule for this JSR: First meeting: July 22, 2003. JCP membership review: July 5, 2004. Public review: January 3, 2005. EC approval: July 1, 2005.

Anticipated working model for the Expert Group: Group will meet via teleconference (approximately monthly), plus occasional personal meetings (approximately 4 per year). The Specification Lead (i.e. The Open Group) will establish the specific rules for Expert Group representation and for reaching decisions.

How will RI and TCK be delivered (part of profile or platform edition, stand-alone, both, including profile or platform version)? Stand-alone.

Rationale of previous versions are stand-alone and this RI and TCK will be delivered as part of a profile or platform edition: Not applicable.

Business terms for Spec, RI, and TCK: Specification will be publicly available. RI and TCK are to be [made available under appropriate licensing terms](#).

Deleted: licensed using an accepted open source licensing model.

Existing documents, specifications, or implementations describing the technology: Andy Wellings' Ravenscar Java paper may influence the Expert Group. Real-Time Core Extensions version 1.0.14. High Integrity Profile version 0.2. Espresso Project API and Implementation Notes (<http://www.irisa.fr/mtl-expresso/docs/hip-api.pdf>). Real-Time Specification for Java version 1.0.1.

As of 23 July 04

How might these documents be used as a starting point for this work? Determined by the Expert Group.

Additional information (if any):

As of 23 July 04