One Pager: <Project/Component Working Name>

Table of Contents

1. Introduction

1.1. Project/Component Working Name

// Name of the Project or Component

1.2. Name(s) and e-mail address of Document Author(s)/Supplier

// The individual who are wrote this document
// Name: email address

1.3. Date of This Document

// MM/DD/YY

2. Project Summary

2.1. Project Description

// A SHORT description of this project suitable for use
// on dashboards and status rollups.
// See below for a longer, more detailed technical description

2.2. Risks and Assumptions

// Note any risks, and assumptions that must be considered along
// with the proposal. Include technical risks.

3. Problem Summary

3.1. Problem Area

// What problem or need does this project solve?

3.2. Justification

// Why is it important to do this project?

4. Technical Description

4.1. Details

// To the extent known, how is this project going to be done?
// This information is used by the reviewer to get a feel for the
// complexity and risk involved, and
// the architectural constraints that this project is working
// under. Try to present alternatives and show relationships to
// existing or proposed projects/standards.

4.2. Bug/RFE Number(s)

// List any Bug(s)/RFE(s) which will be addressed by this proposed change.
// Provide links to the Issue tracker Bug(s)/RFE(s)where possible

4.3. In Scope

// Aspects that are in scope of this proposal if not obvious from above.

4.4. Out of Scope

// Aspects that are out of scope if not obvious from above.

4.5. Interfaces

// Interfaces are a major part of Architectural review.
// Commands, Files, Directory Layout, Ports, DTD/Schema, admin tools,

// config files, APIs, CLIs, and almost anything that is externally
// observable is an interface. In 1-Pager it is necessary to document
// any interface that can be used by external projects and products.
// Documented public interfaces must be assigned a stability level.
// Some commonly used Stability levels in prior projects are:
//
// Stable : Widely used public interface. changed very rarely.

//
// Standard : Defined by a standards body (e.g: JDBCv3). Rare but
// incompatible clarifications and changes could occur
// in a standard. Product will specify version of std
// supported. J2SE, J2EE and WS* are classified
// as Standard.
//
// Evolving : Subject to carefully controlled but possibly
// incompatible change at a major or minor release.
// When a change is made all efforts will be made
// to address incompatiblity and migration. All
// incompatibilities will need to be reviewed
// and approved by as-ccc@sun.com.
//
// Unstable : Early access, subject to unrestricted degree of
// change. A few App Server interfaces are classified
// as Unstable. Docs must call out exported unstable
// interfaces. Be wary of importing Unstable interfaces.
//
// External : Defined external to GlassFish Application Server,
// but not by a Standards body. Suitable for describing
// other freeware, open source interfaces.
//
// http://www.opensolaris.org/os/community/arc/policies/interface-taxonomy/
// describes the permitted interface taxonomy.

4.5.1 Exported Interfaces

// Disclose all interfaces that this project exports.

Interface	Stability	Former Stability (if changing)	Comments

4.5.2 Imported interfaces

// Disclose interfaces this project imports.

Interface	Stability	Exporting Project: Name, Specification or other Link.	Comments

4.5.3 Other interfaces (Optional)

// Any private interfaces that may be of interest?

Interface	Stability	Exporting Project: Name, Specification or other Link.	Comments

4.6. Doc Impact

// List any Documentation (man pages, manuals, service guides...)
// that will be impacted by this proposal.

4.7. Admin/Config Impact

// How will this change impact the administration of the product?
// Identify changes to GUIs, CLI, agents, plugins...

4.8. HA Impact

// What new requirements does this proposal place on the High
// Availability or Clustering aspects of the component?

4.9. I18N/L10N Impactinks

// Does this proposal impact internationalization or
// localization?

4.10. Packaging & Delivery

// What packages, clusters or metaclusters does this proposal
// impact? What is its impact on install/upgrade?

4.11. Security Impact

// How does this proposal interact with security-related APIs
// or interfaces? Does it rely on any Java policy or platform
// user/permissions implication? If the feature exposes any
// new ports, Or any similar communication points which may
// have security implications, note these here.

4.12. Compatibility Impact

// Incompatible changes to interfaces that others expect
// to be stable may cause other parts of application server or
// other dependent products to break.

// Discuss changes to the imported or exported interfaces.

// Describe how an older version of the interface would
// be handled.

// List any requirements on upgrade tool and migration tool.

4.13. Dependencies

// List all dependencies that this proposal has on other
// proposals, components or products. Include interface
// specifics above in the interfaces section;
// LIST dependency component version requirements here.

5. Reference Documents

// List of related documents, if any (BugID's, RFP's, papers, Blogs)

// List of related documents, if any (BugID's, RFP's, papers, Blogs).
// Explain how/where to obtain the documents, and what each
// contains, not just their titles.

6. Schedule

6.1. Projected Availability

// Dates in appropriate precision (quarters, years)