

```
// Delete comments before final submission.  
// make sure that the response fits within 80 columns  
012345678901234567890123456789012345678901234567890123456789012345678901234  
56789
```

One pager template version: 1.9

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 Bug(s)/RFE(s) where possible.

// RFE's must be trackable via an issue in Issue Tracker for

// features in the open source distro and in Bugster for value-add

// features to be released in the commercial distro.

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 may be commands, files, directory structure, ports,
// DTD/Schema, tools, APIs, CLIs, etc.
// Note: In lieu of listing the interfaces in the one pager,
providing
// a link to another specification which defines the interfaces
// is acceptable.

4.5.1. Public Interfaces:

// List new, public interfaces this project exports.

4.5.2. Private Interfaces:

// List private interfaces which are externally observable.

4.5.3. Deprecated/Removed Interfaces:

// List existing public interfaces which will be deprecated or
// removed by this project.

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 Impact:

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

4.10. Packaging, Delivery & Upgrade:

4.10.1. Packaging

// What packages does this proposal impact? How will the
packages
// be impacted? Will new IPS/pkg(5) packages need to be

created?

4.10.2. Delivery

// What impact will this proposal have on product installation?

4.10.3. Upgrade and Migration:

// What impact will this proposal have on product upgrade and/

or

// migration from prior releases? Enumerate requirements this
// project has on upgrade and migration.

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.

4.13. Dependencies:

// An internal dependency is a dependency on a project, module,
// component or product that is within the GlassFish project.
// An external dependency is a dependency on a project, component
// or product that resides outside of the GlassFish project.

4.13.1 Internal Dependencies

// List all internal dependencies this proposal has on other
// software. Include component version requirements if
necessary.

4.13.2 External Dependencies

// List all external dependencies this proposal has on other
// software. Indicate if the software is open source, what
// license terms the software is released under and the version
// of the software that is being consumed by this project.

4.14. Testing Impact

// How will the new feature(s) introduced by this project be
tested?

// Do tests exist from prior releases (e.g. v2) that can be reused?

// Will new tests need to be written? Can they be automated?

5. Reference Documents:

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

// Explain how/where to obtain the documents, and what each

// contains, not just their titles.

6. Schedule:

6.1. Projected Availability:

// Indicate which milestone from the current schedule the project

// will be:

// * Initially integrated (may not be feature complete)

// * Feature complete (ready for handoff to QA)

// * At production quality level