

git_comments:

git_commits:

1. **summary:** Merged revisions 997892 via svnmerge from
message: Merged revisions 997892 via svnmerge from <https://svn.apache.org/repos/asf/cxf/trunk>
r997892 | dkulp | 2010-09-16 15:45:52 -0400 (Thu, 16 Sep 2010) | 2 lines [CXF-2995] Patch use of cxf with lazy inits. Patch from Adam Lewandowski applied git-svn-id:
<https://svn.apache.org/repos/asf/cxf/branches/2.2.x-fixes@997900> 13f79535-47bb-0310-9956-ffa450edef68

github_issues:

github_issues_comments:

github_pulls:

github_pulls_comments:

github_pulls_reviews:

jira_issues:

1. **summary:** Bus configuration ignored if Spring context has default-lazy-init=true
description: Bus configuration using the Spring namespace handler is ignored if the containing context defaults to lazy-init (default-lazy-init=true). The configuration bean definition (BusDefinitionParser\$BusConfig) created by BusDefinitionParser inherits this setting. Since the bean is never explicitly referenced it does not get instantiated which would trigger the setApplicationContext method, registering the configured bus under the 'cxf' bean name. Instead, you get a bus with the default configuration. As a workaround, you can set the containing context to eager-init, but this is not feasible in many cases. I've attached a patch which includes a test case and fix in BusDefinitionParser to force the config bean to be eager-init. I've read a few places that Daniel Kulp mentions most CXF beans will be lazy-init in 2.3, perhaps this will be addressed then. In the meantime, this patch is working for me.
label: test
2. **summary:** Bus configuration ignored if Spring context has default-lazy-init=true
description: Bus configuration using the Spring namespace handler is ignored if the containing context defaults to lazy-init (default-lazy-init=true). The configuration bean definition (BusDefinitionParser\$BusConfig) created by BusDefinitionParser inherits this setting. Since the bean is never explicitly referenced it does not get instantiated which would trigger the setApplicationContext method, registering the configured bus under the 'cxf' bean name. Instead, you get a bus with the default configuration. As a workaround, you can set the containing context to eager-init, but this is not feasible in many cases. I've attached a patch which includes a test case and fix in BusDefinitionParser to force the config bean to be eager-init. I've read a few places that Daniel Kulp mentions most CXF beans will be lazy-init in 2.3, perhaps this will be addressed then. In the meantime, this patch is working for me.
3. **summary:** Bus configuration ignored if Spring context has default-lazy-init=true
description: Bus configuration using the Spring namespace handler is ignored if the containing context defaults to lazy-init (default-lazy-init=true). The configuration bean definition (BusDefinitionParser\$BusConfig) created by BusDefinitionParser inherits this setting. Since the bean is never explicitly referenced it does not get instantiated which would trigger the setApplicationContext method, registering the configured bus under the 'cxf' bean name. Instead, you get a bus with the default configuration. As a workaround, you can set the containing context to eager-init, but this is not feasible in many cases. I've attached a patch which includes a test case and fix in BusDefinitionParser to force the config bean to be eager-init. I've read a few places that Daniel Kulp mentions most CXF beans will be lazy-init in 2.3, perhaps this will be addressed then. In the meantime, this patch is working for me.
label: code-design
4. **summary:** Bus configuration ignored if Spring context has default-lazy-init=true
description: Bus configuration using the Spring namespace handler is ignored if the containing context defaults to lazy-init (default-lazy-init=true). The configuration bean definition

(BusDefinitionParser\$BusConfig) created by BusDefinitionParser inherits this setting. Since the bean is never explicitly referenced it does not get instantiated which would trigger the setApplicationContext method, registering the configured bus under the 'cxf' bean name. Instead, you get a bus with the default configuration. As a workaround, you can set the containing context to eager-init, but this is not feasible in many cases. I've attached a patch which includes a test case and fix in BusDefinitionParser to force the config bean to be eager-init. I've read a few places that Daniel Kulp mentions most CXF beans will be lazy-init in 2.3, perhaps this will be addressed then. In the meantime, this patch is working for me.

jira_issues_comments:

1. **body:** Patch including a test case and fix
label: test
2. Patch applied. Thanks!