3VR Chain Upgrade Design Document

`

Name (s): Jackie.zhang@3vr.com

Lab Section:

Workstation:

Date: (04/28/2017)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Date** | **Version** | **Originator/Reviewer** | **Comments** | **Remarks** |
| 04/28/2016 | 0.1 | Jackie Zhang | Initial Draft |  |
| 05/03/2016 | 0.2 | Chunmei Wu | Review with comments |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

**TABLE OF CONTENTS**

[1. Introduction 1](#_Toc481577376)

[2. System Overview 1](#_Toc481577377)

[3. Detailed Design 1](#_Toc481577378)

[3.1 Overview 1](#_Toc481577379)

[3.2 Code Changes 2](#_Toc481577380)

[3.2.1 SystemManager 2](#_Toc481577381)

[3.2.2 Shell 2](#_Toc481577382)

[4. Error Conditions 3](#_Toc481577383)

[5. Q&A 3](#_Toc481577384)

1. Introduction

This document describes the architecture and system design of 3VR Chain-Upgrade.

In some scenarios, it maybe cause cannot upgrade to newer version. For example:

The appliance current version is 7.2.21, configure some channels, but some channel drivers used by these channels been removed in 8.1.1.1, but added back those channel drivers in 8.1.1.10, so the user cannot upgrade this appliance to 8.1.1.1, and also cannot upgrade to 8.1.1.10.

In current upgrade logic, it cost more times and it is cockamamie. For example:

If user want to update one appliance from 7.3.2 to 8.1.1.10, it has to update this appliance to 8.1.1.1 with hotfix package, after this update finish, then have to update this appliance to 8.1.1.10 with path package.

So, decided to provide Chain-Upgrade feature to reduce issues and times.

1. System Overview

Chain upgrade allows customer to apply the hotfix and patch at the same time.

Following are the deployment requirements:

* Only allow one hotfix together with one patch

We don’t support multiple hotfixes with one patch.

* Doesn’t support scheduled upgrade
* Only support this feature by upgrading from Calistoga release to newer version.

We don’t support the chain upgrade from 7.x or 8.x to Calistoga release.

1. Detailed Design
   1. Overview

Normally, we could just release one upgrade package by merge hotfix package and patch package. But it will cost more related works (like release note, and other related docs, etc….) and spent more times to test this package although the QA already test hotfix package and patch package.

So, to reduce upper issues, we have to do it by codes.

When user do chain-upgrade, they have to send only one hotfix package and only one patch package, and we do below steps to do chain-upgrade:

1. First, check whether the hotfix package and patch package are suitable. For example, we can’t do chain upgrade using 8.x hotfix together with 9.x patch.
2. Second, if they are suitable, replace the 01-Build.zip of hotfix package by patch package files, otherwise, let the user know the hotfix package and patch package not suitable.
3. Third, do upgrade if they are suitable.

1). First upgrade by hotfix package.

2). Second upgrade by patch package.

* 1. Code Changes
     1. SystemManager

Add a new button “Chain Upgrade”, after user click “Chain Upgrade” button, show a dialog to let user choose upgrade packages.

Note: maybe we have to create a new dialog, which could select more than one file, or use OpenFileDialog provided by .Net Framework. But the theme of OpenFileDialog is different with our customized theme.

UI Image

After user choose upgrade package over, check whether user chooses 2 upgrade package files (.3pa extension), if it is not, tell user have to choose 2 upgrade package files (one is hotfix package, the other one is patch package).

Then use service call (maybe new service call to avoid cannot send 2 upgrade files to server) to upload those 2 upgrade packages to server.

Then SystemManager call Progress service call to get the status of server.

If the two files not correct, let the user know it.

If the two files not suitable (patch upgrade package is not suit hotfix package), let the user know it.

* + 1. Shell

Maybe need add a new service call to get uploaded files and avoid server just can accept only one upgrade package. (because after upload one upgrade package file, the server will change it is status to Pending, in this status, the client cannot upload another upgrade package file to server).

After get those 2 upgrade package files over,

First, check whether one package is hotfix upgrade package, and the other one is patch upgrade package. If not, change the progress xml to tell the Client (SystemManager), these 2 files are incorrect. They have to upload hotfix package and patch package.

Second, if one package is hotfix package, and the other one is patch package, check whether the hotfix package and patch package are suitable (BaseHotfixVersion setting of patch package is same as hotfix package version). If not, change the progress xml to tell client (SystemManager) those 2 files are incorrect. The patch package is not suitable with hotfix package.

Third, if hotfix package and patch package are suitable,

1. Replace 01-Build.zip of hotfix package with patch package files.
2. Upgrade by hotfix package.
3. Upgrade by patch package.
4. Error Conditions

* For Chain-Upgrade, only one hotfix package and one suitable patch package which base package is same as the hotfix package can accepted. If the hotfix package and the path package is not suitable, will show detail error message to the user to let user know the hotfix package and patch package is not suitable.
* aa

1. Q&A

* Hot to determine hotfix package and patch package?
  + First check whether package include UpgradeCheck.zip file, if yes, we can assume this package file is upgrade package.
  + If also include 01-Build.zip, we can assume this package is hotfix upgrade package.
  + If not include 01-Build.zip, we can assume this package is patch package.
* How to determine whether the hotfix package and package are suitable?
  + First, get the hotfix package version from BuildNumber.txt file which store at UpgradeCheck.zip.
  + Second, get the base hotfix version of patch package from <BaseHotfixVersion> element of upgradeInfo.xml.

Note: before path release, the release manager has to add or modify the <BaseHotfixVersion> element value of upgradeInfo.xml of patch package.

* + Then check whether the major version between hotfix package and patch package are same. If yes, we can assume the hotfix package and patch package are suitable. Otherwise, they are not suitable.
  + For example: the hotfix package version is 8.1.1.2.2005, and the base hotfix version of patch package is 8.1.1, so the major version between hotfix package and patch package is same, both are 8.1.1, we assume they are suitable, otherwise not suitable.
* Why we just do upgrade check hotfix package when do chain-upgrade?
  + Because we can assume the patch package always can upgrade from the base version (hotfix package version).
  + So, we just check whether could upgrade by the hotfix package. If yes, we can assume could upgrade current version to hotfix package version, and also could upgrade to patch version from hotfix version.
* Why first upgrade by hotfix package and then upgrade by patch package, although already replaced 01-Build.zip of hotfix package by patch package files?
  + Maybe the patch package will do some special works. So, to reduce risks, we have to upgrade by patch package.
* aaa