



Content Portal Patch Guidance

Version 6.1.x

Table of Contents:

Table of Contents:	1
Document Purpose:	1
Git repository of Front end templates:	1
Customer Development process:	1
Patching Prerequisites:	3
Required Packages:	3
Upgrade Process:	3
How to install each package with Meteor:	4
How to manually add the Jadu Rescheduler (one time only):	4
How to clear cache:	4
How to verify the upgrade:	5
Patch Migrations & Troubleshooting:	6
Notable changes in Content Portal 6.1	6
Hyland OnBase integration	6
SAML User Adapter	7

Document Purpose:

This document is aimed at assisting Systems Administrators in the application of their first Jadu Content Portal upgrade patch. Some basic information is also provided for any web developers wishing to extend the platform and how these should be managed through version controlled.

Git repository of Front end templates:

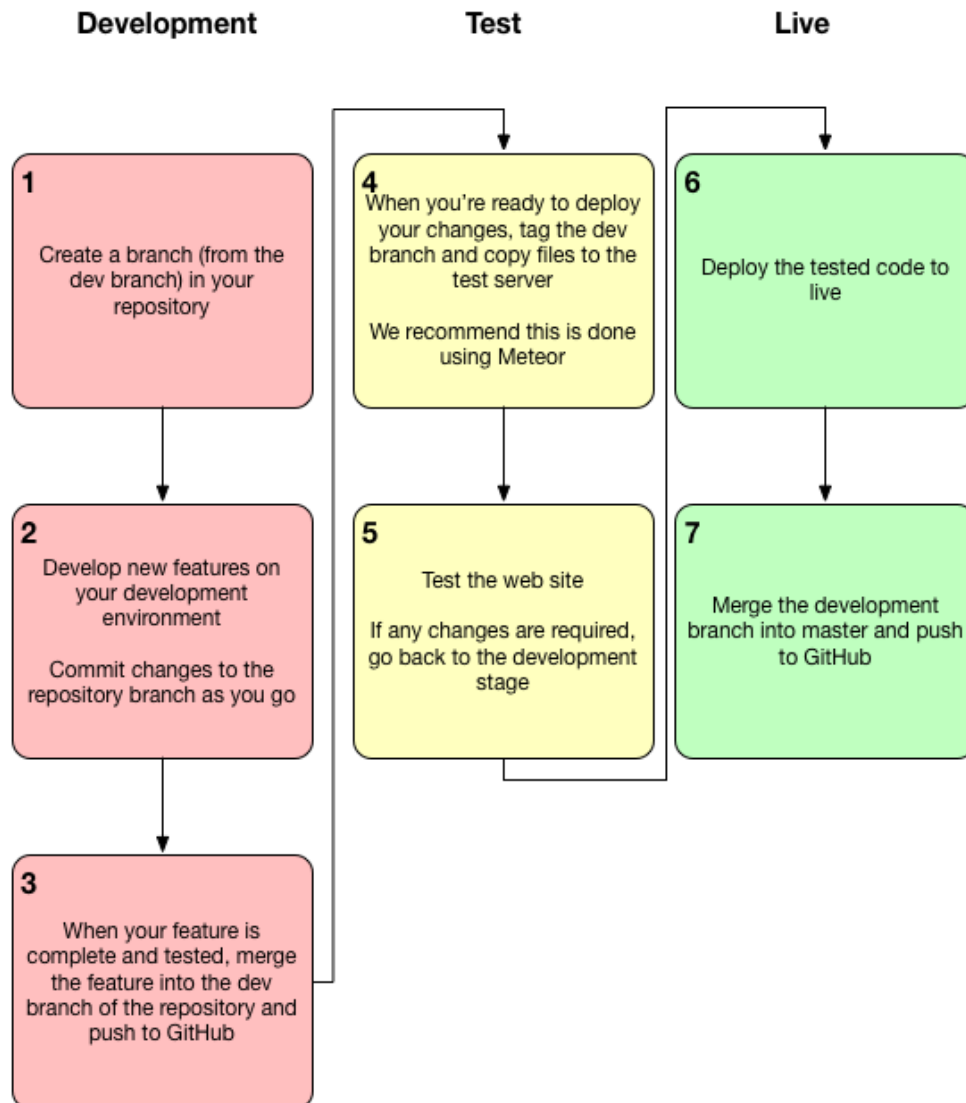
Some customers wish to make changes themselves or use an independent third party. We have an established approach that ensures any changes made by customers or their nominated third parties are done so in a controlled way that all parties can collaborate on a single codebase where necessary. This is achieved using [GitHub](#).

Customer Development process:

Start off by committing the standard provided Front end templates as supplied by Jadu to your project in a 'develop' branch.

We recommend that any changes required to standard templates or to be added to respective “custom” folders are made in branches taken from the ‘develop’ branch and that only tested, deployable changes are then merged back into ‘develop’. This will ensure that only working code is ever deployed to live, repeating as necessary for any features / changes being introduced. A ‘master’ repository is then updated from ‘develop’ once deployments have been successfully released into PROD.

The following flowchart presents a simple development workflow using a ‘feature branching model’ that we recommend is used either when Jadu provide platform management or when this is carried out by the customer.



Future Content Portal template patch releases can then be diff'd against your repository to merge in all relevant future updates alongside your already committed customizations allowing you to assess any impact and allow you to focus areas of UAT etc as part of patch releases.

Note: At the present time, if customers perform their own platform management, including the application of patches, and you use the Composer dependency manager, then Jadu would then need to produce a package each time a new deployment to the test environment was required (that same package, if it passes testing would then be deployed to PROD). The creation of a release package in this way may incur additional support charges, so please discuss this with us first if you plan to introduce the use of Composer.

Patching Prerequisites:

1. Verify that the DB version is MariaDB 10.1 (LAMP only)
2. Verify that the PHP version installed is PHP 7.2 (minimum 7.2.5)
3. You have patched to at least CP 5.2 before applying this patch
4. Take a backup of config/perceptive/constants.xml
5. Take a backup of the /public_html/site/ folder (as the web development team may have customised some of these front end template scripts - see Git repository section of this document)
6. Download the packages listed in the Required packages section below and upload to your web server(s).

Note: It is advisable that you manually add the Jadu Rescheduler task after installation & setup - this will only need to be done once if not already configured - steps have been provided in a later section for this.

Required Packages:

1. **JCP-6.1.x.zip** = JCP bundles with XFP and CMS
2. **JCP-6.1.x-Templates.zip** = Full JCP Front end template bundle

Upgrade Process:

We would recommend that this process be followed on your UAT environment first with a round of testing prior to application to PROD if currently actively using PROD at this time. Each step mentioned below is described in more detail afterwards:

1. Turn off forms via XForms Pro > Settings in the control centre
2. **WISP Only:** Stop IIS.
3. Install JCP 6.1.x template package via Meteor (see below steps)
4. Install JCP 6.1.x core package via Meteor (same process as above)
5. Manually clear cache (although Meteor does also clear caches)
6. Warm caches using the following 2 CLI commands:

LAMP

```
su jadu-www -s /bin/bash -c "/usr/bin/php /var/www/jadu/cli.php cache:warmup --kernel=frontend"  
su jadu-www -s /bin/bash -c "/usr/bin/php /var/www/jadu/cli.php cache:warmup --kernel=cc"
```

WISP (via command prompt, run from the jadu install directory e.g. C:\inetput\wwwroot\jadu)

```
php cli.php cache:warmup --kernel=frontend  
php cli.php cache:warmup --kernel=cc
```

7. **WISP Only:** Start IIS
8. Turn on forms via XForms Pro > Settings in the control centre
9. Manually Verify the upgrade / connectivity of the system
10. Reapply your front end templates (where customisations have taken place - ideally from version control)

Note: Front end template files will be replaced as part of applying step 3 above. Your team may therefore need to reapply manually any front end customisations previously made as a final step, as these changes will have been overwritten during the patching process. See Git Version control section for how you version control such changes better.

How to install each package with Meteor:

Patches to the Jadu software are applied through our Open Source Meteor tool. Further details on the various capabilities of this tool are covered in more detail at: <http://meteor.jadu.net/guides/patch-apply/>.

In Multi-server / load balanced environments, patches must be applied to all web nodes that are part of that environment's cluster. Meteor will only run database migrations on the first node and no database migrations are run on subsequent nodes during the patching process.

You will need to repeat this process for each patch package listed previously and in the specified order.

1. Copy and unzip the package to the jadu root folder.
2. From the terminal, cd to the unzipped folder, run the following command and answer yes to run the migrations.

```
php -d memory_limit=512M meteor.phar patch:apply --path=<<JADU-HOME>>
```

Should you ever need to revert from the patch, from inside the patch directory run the following command and then clear the system cache:

```
php -d memory_limit=512M meteor.phar patch:rollback
```

How to manually add the Jadu Rescheduler (one time only):

On ONE of your web nodes (ONLY) use the Windows Task Scheduler to add a daily task for a quiet local time in the day (e.g. 04:00am), to run the following command (where "F:\inetpub\wwwroot\jadu" is your root Jadu installation directory):

```
F:\inetpub\wwwroot\jadu\var\scheduled_tasks\run_rescheduler.bat
```

Note: Please check that the path to the PHP executable is correct within the above .bat file when setting up this process.

How to clear cache:

CentOS System:

1. Clear the Jadu Cache via the /jadu/maintenance/cacheBash.php page in the Jadu Control Centre.
2. In a terminal running as an administrator user on a web node, run the following to warm the cache again:

```
php -d memory_limit=-1 /var/www/jadu/bin/warm-cache cc  
php -d memory_limit=-1 /var/www/jadu/bin/warm-cache frontend
```

3. Then run the command to restart httpd

service httpd restart

Windows System:

1. Clear the Jadu Cache via the /jadu/maintenance/cacheBash.php page in the Jadu Control Centre
2. In a terminal running as an administrator user on the database node, run the following
 - a. If using redis:

```
redis-cli.exe flushall  
(you may need to specify the full path to redis-cli.exe)
```

- b. If using memurai:

```
memurai-cli.exe flushall  
(you may need to specify the full path to memurai-cli.exe)
```

3. In a terminal running as an administrator user on a web node, run the following to warm the cache again (where "F:\inetpub\wwwroot\jadu" is your root Jadu installation directory):

```
php -d memory_limit=-1 F:\inetpub\wwwroot\bin\warm-cache cc  
php -d memory_limit=-1 F:\inetpub\wwwroot\bin\warm-cache frontend
```

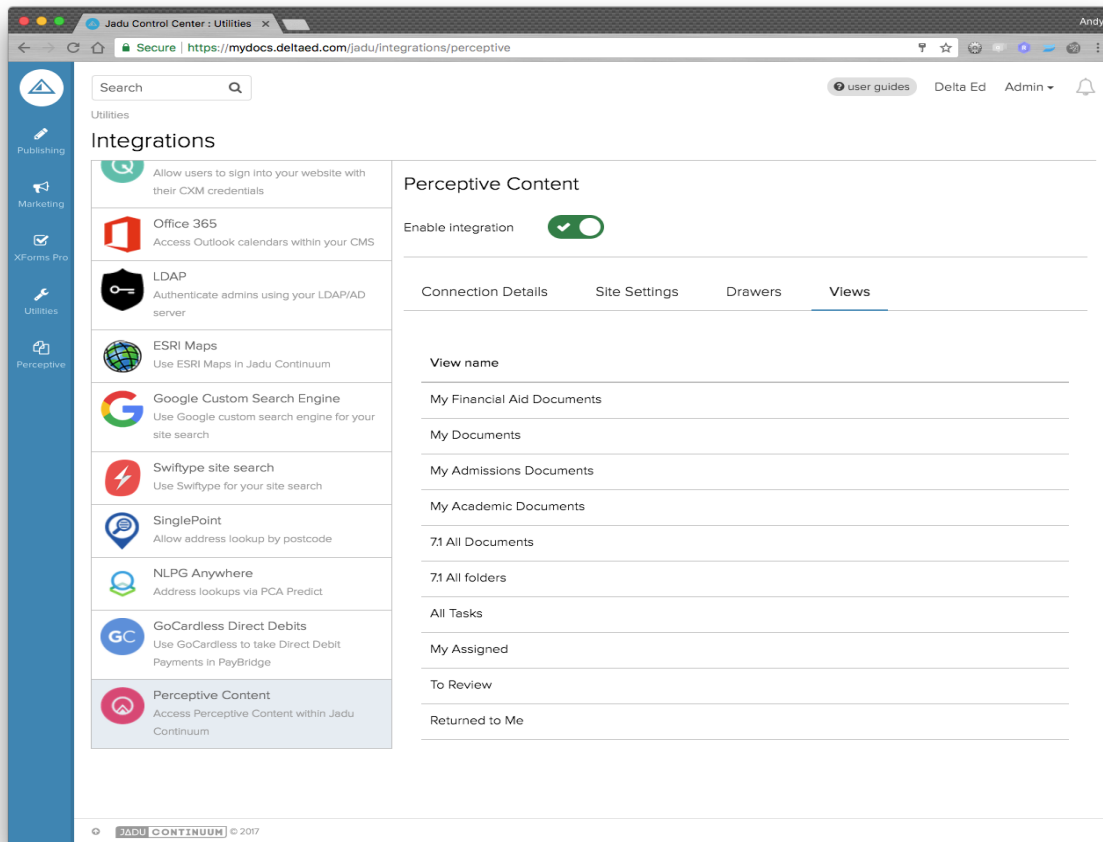
4. In a terminal running on each web node, run the following to restart IIS:

```
iisreset
```

How to verify the upgrade:

Where Perceptive Content / ImageNow integration has been in use prior to patch application, it would be best to check that connectivity has not been impacted by this upgrade.

This can be done by logging into the Jadu Control Centre, going to the Utilities > Integrations page and selecting "Perceptive Content" from the list of systems on the left. If all configuration is still in place, then you will be able to click on the "Drawers" or "Views" tabs in order to see a list of these resources displayed (permissions permitting to the nominated system user account). This will show that connectivity between the systems is still active.



Patch Migrations & Troubleshooting:

If there are existing forms in the system with identical question names when applying a patch to an existing environment, then the XFP migrations may report an issue. This requires resolving in the database before then resuming the patch. Contact us if you hit this particular issue when trying to apply this patch and we'll try to assist you in how to resolve this.

Notable changes in Content Portal 6.1

Hyland OnBase integration

Integration with Hyland OnBase has been added in this release. Further information on the functionality available and how this works can be found in the [Content Portal Implementation Guide](#).

One of the elements of this integration is a widget which requires additional js and css to be included via the templates. If you are using the standard template set, this change will be included via the patch. If you have a customized version of the templates and wish to use the OnBase integration, you will need to update your templates manually with this change, shown in the diff below.

public_html/site/includes/stylesheets.php

```
<?php
    if (isset($homepage) && $homepage !== null) {
?>
+     <link rel="stylesheet" type="text/css" href="<?php print
getStaticContentRootURL();?>/site/styles/widgets/onbase_datatable.min.css">
+     <link rel="stylesheet" type="text/css" href="<?php print
getStaticContentRootURL();?>/site/styles/widgets/onbase.css">
        <link rel="stylesheet" type="text/css" href="<?php print
getURLToWidgetStylesFileForSeason($STYLESHEET); ?>">
<?php
    }
```

SAML User Adapter

In previous versions of the SAML user adapter, there was an issue where if the “User Identifier Attribute” field (on the Matching Attributes tab of adapter settings) was non-empty, this value would get ignored, and the value of the configured “User Email Attribute” would be used instead. Additionally, the “External ID prefix”, if left empty, would default to “SAML_”. This would result in JaduUser records with the external ID in the format “SAML_<email address>”. This caused problems with integrations which required the real IdP identifier (e.g. AD username) to be used when sending or surface data.

As when the user signs in for the second and any subsequent times, the external ID value is used to match with an existing JaduUser record, changing the way this is built up for existing implementations would cause new user records to be created for any existing users the next time they signed in (following upgrade to CP 6.1 or higher). This would mean that user’s would no longer be able to access any data linked to their previous user record.

As it is not possible for us to reliably automate the fix of existing data, for implementations already using the SAML adapter prior to this upgrade, we have included a migration which will ensure that users continue to be identified in the same way. After applying the patch, you will notice therefore that the settings on the Matching Attributes tab will be as follows:

- User Email Attribute - value as it was pre-patch
- Use email address as user identifier - toggle enabled
- External ID prefix - “SAML_”

The issue in the software has been fixed so that if the “Use email address as user identifier” toggle is disabled, you will have the option to enter the “User Identifier Attribute”, the value of which is now used correctly. You can also remove the “External ID prefix” value, which will result in no prefix being used.

If no existing user data exists in the system (or it only contains test data), it’s safe to change these so that new user records going forwards would have their external ID set correctly. If you have existing users and require the new behaviour, you would need to undertake a data migration / cleansing exercise to correct the external IDs of existing JaduUser records, and any other systems to which the previous value may have been passed.

JADU

EVERYTHING / POSSIBLE