In this demonstration, you will see how easy it is to apply WebSphere Middleware maintenance to both patterns and pattern instances using the integrated Installation Manager Repository.

First, there will be a review of the structure of the Installation Manager Repository to see how the software versions and fixes are organized. Then, there will be a review of the components inside the pattern builder that allow for the application of maintenance fixes and fix packs at deployment time.

There are two instances of a pattern on this system that are running the trade-like application. The pattern includes a single WebSphere node and a single DB2 node. DB2 does not use installation manager for its maintenance function. However, DB2 is an important part of many patterns. So applying maintenance to DB2 is an important part of any maintenance effort.

One of the pattern instances was deployed before the latest DB2 fix pack was loaded onto the system, and so it is a candidate for an upgrade. That will be our first task—to upgrade that DB2 node to the latest version. The other pattern instance will have maintenance applied to its WebSphere node: both emergency fixes and a more current WebSphere fix pack. And finally, there will be a review of a multinode WebSphere pattern to see how maintenance is applied to the various nodes in a WebSphere cell.

**Review content of the Installation Manager Repository**
Let’s start this demonstration by taking a look at the Installation Manager Repository and how the various components of it are organized.
Go to the **System** menu and under **Settings**, there is a link to the **IBM Installation Manager Repository**.

This system has a number of categories inside the Installation Manager Repository. Shown here is the Liberty_Install_Software category and the WebSphere category.

Expanding **WebSphere** shows two emergency fixes that have been loaded at the top of the hierarchy. There are two entries for IBM HTTP Server [IHS]. One contains the 8.5.5.4 fix pack for the 8.5 release level, and the other includes the 8.0.0.10 for the 8.0 release level. Similarly, the Network Deployment section at the bottom of the page lists the 8.5.5.4 fix pack for the 8.5 release.

Expanding the **8.5.5.4 fix pack** underneath **Network Deployment** displays two additional emergency fixes. These two fixes have more restrictive dependencies than the first two that were displayed and have listed themselves under the 8.5.5.4 fixpack level.

This repository also includes a category named **WebSphereDefaultRepo**. This is where the 8.5.5.2 fix pack has been installed for the IBM HTTP Server component.

And finally there is a BPM category. It includes, for Network Deployment, the 8.5.5.3 and 8.5.5.5 fix packs along with some associated emergency fixes.

At first glance, this is a bit confusing, having multiple WebSphere categories. However, it will be a good example for showing how the maintenance functions later on use the content of the repository more than the visual structure of it to find appropriate fix packs and emergency fixes.

As mentioned before, DB2 does not use the Installation Manager. To get to its fix packs, go to the **Catalog** menu and then to **DB2 fix packs for virtual systems**. The list of DB2 fix packs includes one entry, providing the 10.5.0.6 database level.
**Identify the maintenance pattern components**

Next, we will take a look at the maintenance components inside the pattern builder that allow the specification of individual middleware versions and the application of specific emergency fixes.

Let’s take a look at the first sample pattern by going to the Patterns > Virtual System Patterns page. There, I can filter for my pattern name in order to make it easier to find it in the list of patterns.

The sample pattern includes two nodes. The Database node runs the DB2 software component. It uses its own set of maintenance fix packs. The WebSphere node runs one of the WebSphere software components, referencing the Installation Manager Repository for its list of fix packs.

The pattern shows that the drop-down field for the WebSphere fix pack version lists all of the fix packs available in the repository, regardless of which category they were placed in. For this demonstration, the pattern was deployed at the 8.5.5.3 level. Normally, as new versions are loaded into the repository, patterns would be modified to use the most current version.

The DB2 server component uses a specific technique to specify fixpack versions. Attributes on the component provide the various options. The Database version and Installation level are determined by the DB2 blue pattern types that have been installed on the system. The content for the Available fix packs field comes from the list of DB2 fix packs that was just reviewed. Note that, if a fix pack is available, it will be applied at deployment time, so it is not possible to deploy a back-level DB2 node if fix packs are available.

If there is only one version provided by the DB2 pattern types loaded on the system, the version attributes value will be fixed. The installation level may show a varying number of options, depending on how many versions of the DB2 pattern type are currently loaded on the system.
The Fixpack field displays the files in the catalog of DB2 fix packs that was just reviewed. If one exists, it will be automatically selected as shown in this pattern.

There is an interim fix policy that can be added to any node in the pattern. The fixes listed are based on a combination of the OS level and the WebSphere component level. For this pattern, there are two WebSphere fixes that apply to the 8.5.5.3 level. One has been selected, and the other is still displayed in the drop-down field. If the OS level was less than the 2.1.2 version being used here, there may be OS fixes that would be available. It would all depend on which emergency fixes had been loaded onto the system. Applying the policy to the DB2 node would only show appropriate OS fixes as there is not a WebSphere component on the node.

**Review current DB2 and WebSphere instance versions**

Now that we've reviewed the options inside the pattern, let's take a look at what has been deployed and see which middleware can be updated.

The first sample deployment is running the latest version of DB2 and the 8.5.5.3 version of WebSphere. So let's go take a look at the two nodes and verify those versions.

On the WebSphere node, let's run the standard `versioninfo` script to check the version level. And as the pattern specified, 8.5.5.3 is the running WebSphere level. Recall that the pattern also displayed two higher level WebSphere versions available in the Installation Manager Repository. So that provides an opportunity to upgrade this node.

Reviewing the DB2 node of this instance shows it running the 10.5.0.6 level. The pattern actually specified 10.5.0.4 installation level. But the available fix pack was automatically applied to bring it up to the 10.5.0.6 level.

The other sample deployment is running the back-level version of DB2 since it was deployed before the DB2 fix pack was loaded onto the system.
Update the DB2 instance with the latest fix pack
Now let's take a look at upgrading that DB2 node to the latest DB2 fix pack.

This is the sample deployment with the back-level DB2 node. The first step is to place the instance into maintenance mode by selecting the Maintain button at the top of the page. This disables any auto-recovery or auto-scaling operations that might be initiated during the upgrade process. Then, a snapshot of the virtual machine should be taken in the event the upgrade needs to be rolled back so the virtual machine can be restored to its original state.

The Status icon of the instance will reflect the maintenance state. After the upgrade, normal operations will be resumed with the Resume button.

To access the maintenance functions, click the Manage button at the top of the page.

The initial view is the Monitoring page. Click the Operations button at the top of the page to access the Maintenance functions. Note that Maintenance mode can also be started in here by moving the Maintenance mode slider to the right as shown at the top of the page.

The DB2 component in the pattern provides its own maintenance operation tied to an individual node. Click the link on the left to display the maintenance operations for that DB2 node.

Similar to what was displayed in the Pattern Builder, the maintenance operation provides a field with the latest DB2 fix packs that have been loaded onto the system. Select the fix pack, and then click the Submit button to start the process.

The [Operation Execution] Results table at the bottom of the page shows the progress of the operation. The Result status will change as the application of the fix pack goes through its various stages,
from Pending to In Progress and finally to Success. The Return Value column will display a bit more detail on the specific fix pack that has been applied.

Back on the main instance page, normal operations will be resumed by clicking the **Resume** button. If a snapshot was created, it can be deleted once the update has been verified.

Logging back onto the virtual machine and rerunning the DB2 version command shows the update to 10.5.0.6. This is only a high-level verification of the update so additional steps should be taken to ensure that the databases on the system and the application using them are still functioning as expected.

**Update the WebSphere instance with the emergency fix**

Now that the DB2 node in the first sample deployment is updated, it’s time to apply updates to the WebSphere node in the other sample deployment.

As before, with the DB2 node, the instance is first moved to maintenance mode.

On the Operations page, click the **Maintenance** operation to display the two fix options. Recall that the interim fix policy on the WebSphere node in the pattern selected one of two available emergency fixes. The selected fix was applied during the deployment of the node and now shows as being applied to the WebSphere node. The selection field shows the additional fix that can be applied, and it will now be selected.

Selected fixes are added to the list of fixes with the ability to remove them from the list with the **Action** icon. Then the **Submit** button will be selected to start the process.

Similar to the DB2 maintenance operation, the Result column will show the various stages that the maintenance process goes through.
Notice that the DB2 node is included in the results even though there is no WebSphere component on the node. The maintenance operation does not evaluate the nodes before it starts to apply the selected fix. Once the process starts, it will discover that there is no candidate software on the node. This can be seen in the Result details where it shows that no fixes were applicable to the DB2 node, and so none were applied.

And once the fix is successfully applied, it is now included in a list of all fixes that have been applied to this instance.

**Update the WebSphere instance with the WebSphere fix pack**

Now that an emergency fix has been applied, it's time to apply a new WebSphere fixpack version.

Open the **Maintenance fixpacks** section to see what has already been applied. This shows the 8.5.5.3 level that was selected on the WebSphere node component in the pattern. There are two packages in the Installation Manager Repository that are applicable to this level: 8.5.5.4 and 8.5.5.5. For this demonstration, the 8.5.5.4 version will be selected.

And after the fix pack has been successfully applied, it will be displayed in the [Operation Execution] Results table.

It may take a couple of minutes for the console page to be updated, but eventually it will show the new value of the current fix pack.

Now that a new WebSphere level is deployed on the system, there may be emergency fixes that are applicable to this new fixpack level as displayed in the Maintenance fixes list.

Before resuming normal operations, notice that a snapshot has been automatically generated by the maintenance operation. Correct functioning of the WebSphere cell should be verified before the snapshot is deleted.
Switching to the virtual machine command line, the WebSphere version command confirms the successful application of the new fix pack.

That example was fairly simple, with only one WebSphere node. A more realistic deployment will have multiple nodes in the WebSphere cell. So how does the maintenance operation work in this context?

For this pattern, going to the Maintenance Operations page displays a fuller set of fixpack options. The various WebSphere components that are deployed in the pattern instance are displayed in the applied list. It will show the software packages that were applied to each type of node.

In the bottom list of packages that can be applied, the packages are listed by themselves without regard to which node they are running on. Note that only those packages with updates in the Installation Manager Repository are listed. As fix packs are applied, this list will continue to shrink until all fix packs are applied and the list is empty. For this demonstration, the 8.5.5.4 fix pack for Network Deployment has been selected first. It will be applied to all the nodes in the cell that had deployed that software package. Only one of these fix packs can be selected at one time so it makes sense to start with the Network Deployment package and then apply the other packages in turn.

As noted earlier when other fixes were being applied, the maintenance operation doesn't know the topology of the pattern before it starts. But it does extract that topology when the update starts. This means that it will intelligently apply the update to the deployment manager node first before updating the custom and IHS server nodes.

Now that the Network Deployment 8.5.5.4 package has been applied to every applicable node in the cell, additional fix packs can now be applied, which in this example would be the IHS node. Notice that, in this first pass, the IHS node had an emergency fix applied to it even though it was not part of the package update. This demonstrates how maintenance content can be discovered in the Installation Manager Repository. That specific fix was listed under the 8.5.5.3 package entry and so was considered to be a candidate that could be applied to this node concurrently with the other 8.5.5.4 package updates.
These examples demonstrate how the integrated nature of the maintenance operation makes it relatively easy to maintain the version lifecycle of even a large, complex WebSphere cell in only a few steps. Patterns can be easily updated to the latest software versions, and deployed cells can be upgraded in an orderly and structured way with full roll-back capabilities in place.

Copyright © 2016 IBM Corp