

IBM PureApplication System

Creating virtual system patterns



This presentation covers how to create the virtual system patterns in IBM PureApplication™ System.

Table of contents

- Virtual system pattern creation
 - Pattern ordering
- Preloaded virtual system patterns
- Summary

This presentation covers different ways to create virtual system patterns, how to use the editor, how to order the deployments, and advanced functions available on certain virtual image parts.

Creating virtual system patterns

This section will cover how to create virtual system patterns.

Creating virtual system patterns (1 of 2)

- Access virtual system patterns from **Workload Console** → **Patterns** → **Virtual Systems**
 - Quick links on the Welcome page of the Workload console
- **Three ways to create virtual system patterns**
 1. From scratch, using parts in the catalog, and dragging and dropping on the pattern editor
 2. From cloning an existing virtual system pattern
 3. Importing a virtual system pattern from a previously exported pattern from the same or different PureApplication Systems
 - Using command line interface and sample scripts - `exportPattern.py` and `importPatterns.py`



4

Creating virtual system patterns

© 2013 IBM Corporation


All virtual system patterns operations are managed from the patterns page. There are two ways to navigate to that page; either from the “Patterns” menu on the top banner, or from the main welcome page, as shown here.

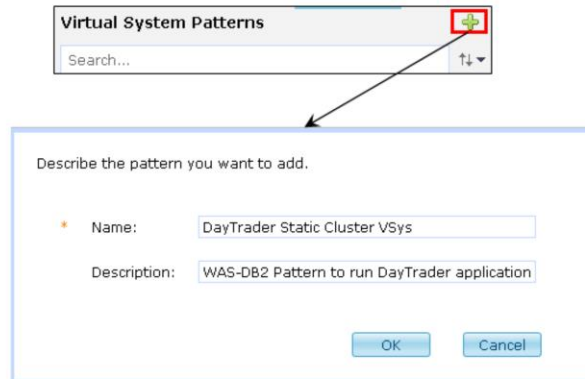
There are three ways to create new virtual system patterns. You can create a new pattern from scratch using the pattern editor and dragging parts from the catalog, like virtual images, script packages, and add-ons.

Another way is to clone an existing pattern. This will allow you to modify or extend existing patterns. IBM-provided patterns for a given topology, representing best practices, come as locked patterns. You can use those patterns by cloning and extending the new pattern.

The third way is importing a previously exported pattern using command line options. The CLI toolkit provides sample scripts for export and import of the patterns. This allows for migrating virtual system patterns between PureApplication Systems.

Creating virtual system patterns (2 of 2)

- From the **Patterns > Virtual systems** page, click the green plus  to create a new pattern, and provide a unique name for the pattern with optional description



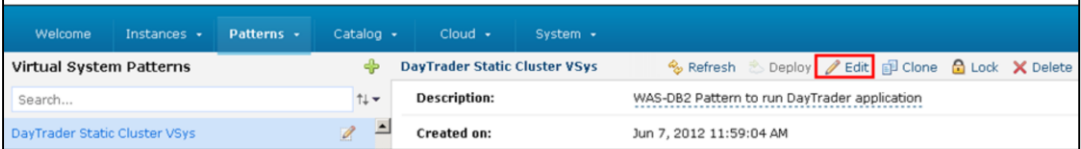
The screenshot shows the 'Virtual System Patterns' interface. At the top, there is a search bar with the text 'Search...' and a green plus icon in a red box. Below the search bar is a modal dialog titled 'Describe the pattern you want to add.' The dialog contains two input fields: 'Name:' with the value 'DayTrader Static Cluster VSys' and 'Description:' with the value 'WAS-DB2 Pattern to run DayTrader application'. At the bottom of the dialog are 'OK' and 'Cancel' buttons.

- Alternatively, you can create a new pattern by cloning an existing pattern (described later in this presentation)

A new pattern can be created from scratch or by cloning an existing pattern. To create a new pattern from scratch, click the green plus sign and provide the unique name for the pattern along with optional description.

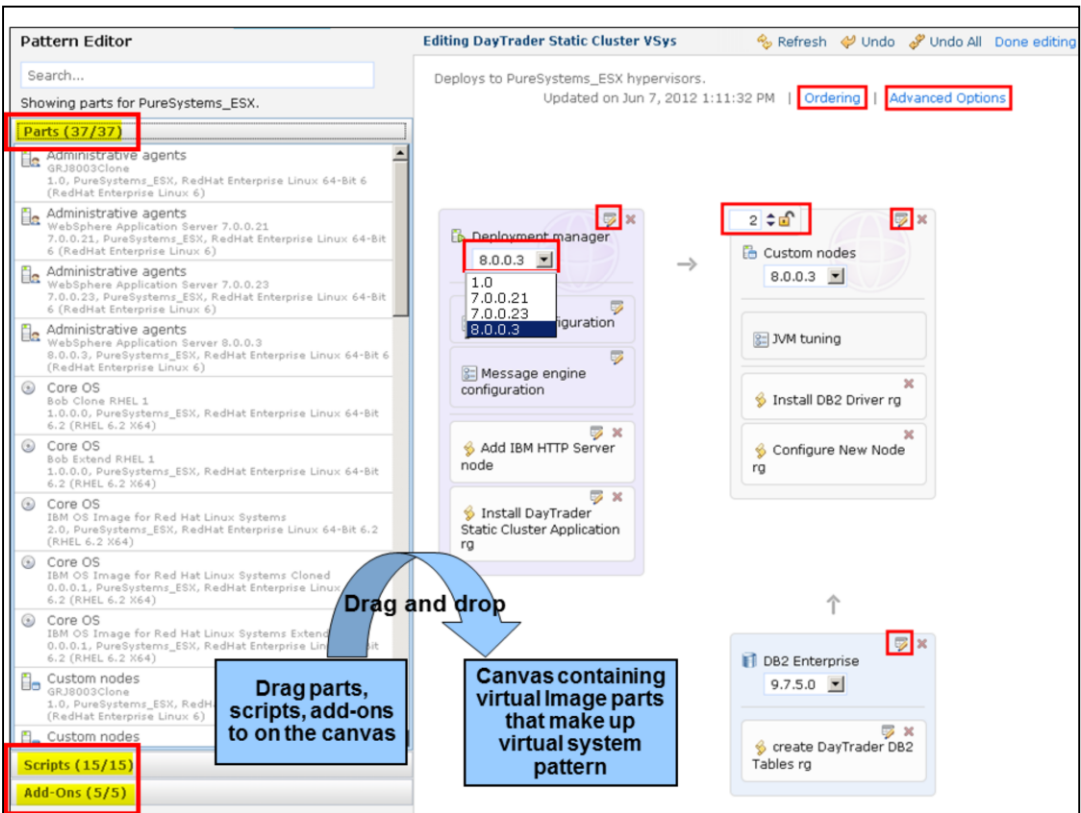
Edit the pattern

- Click the **Edit** function to open the pattern editor.

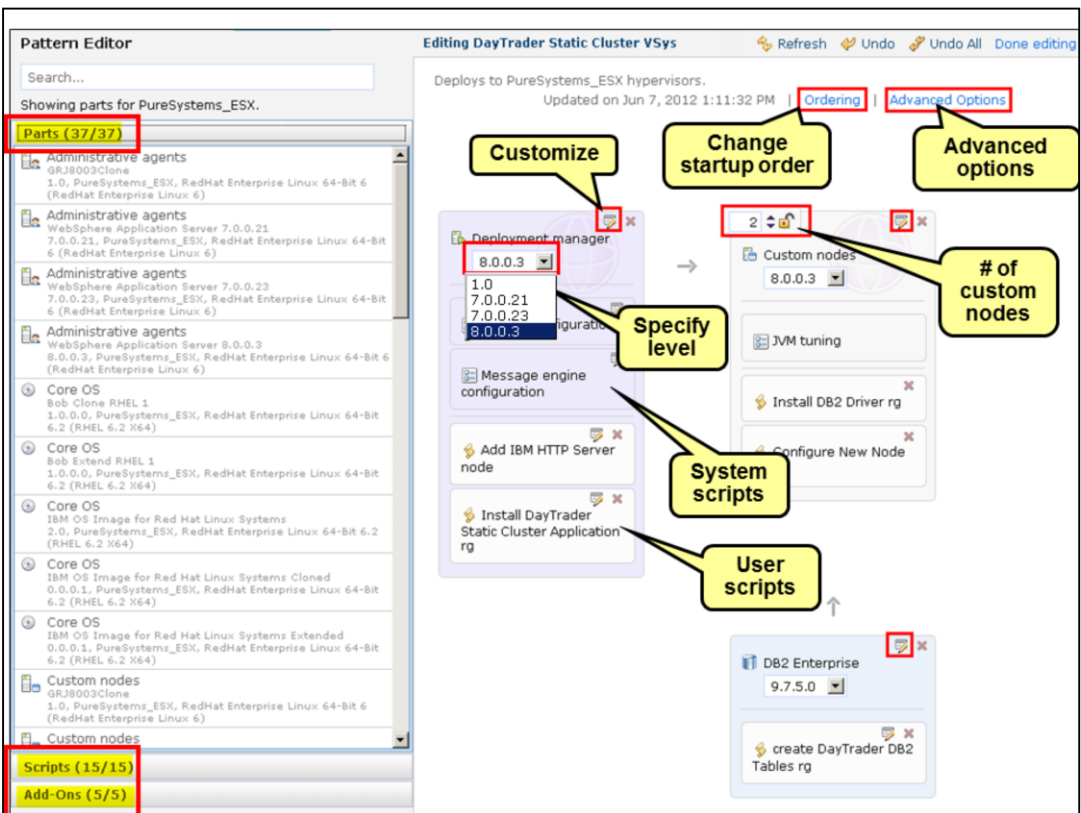


- This opens a pattern editor to create the pattern editor to create the pattern (shown on the next page)

After the pattern is created, click the pencil “**Edit**” icon to edit the pattern. This opens the virtual system pattern editor.



If this is a new pattern to be built from scratch, you are presented with a blank canvas on the right and a palette of virtual image parts, script packages, and add-ons on the left. You create a pattern by dragging parts from the palette on the left and dropping them onto the canvas on the right. In order to add multiple parts at the same configuration point where applicable, like in a custom node, you increment the number located in the upper left corner of the part.



Each part or script can be configured for parameters needed during deployment. Examples include the root password for the operating system, user ID and password for WebSphere administration, and so on.

You can choose the release level and the service level of the virtual image part in the pattern. For example, under “Deployment manager”, if you want to specify release 7.0.0.3, select that level from the pull down.

Click “Ordering” in the canvas to change the sequence of when the different parts are deployed and when the scripts run. This ensures that the right dependency order of startup of scripts can be handled. For example, a DB2 JDBC driver needs to be installed and a database needs to be created before installing the application that accesses the database.

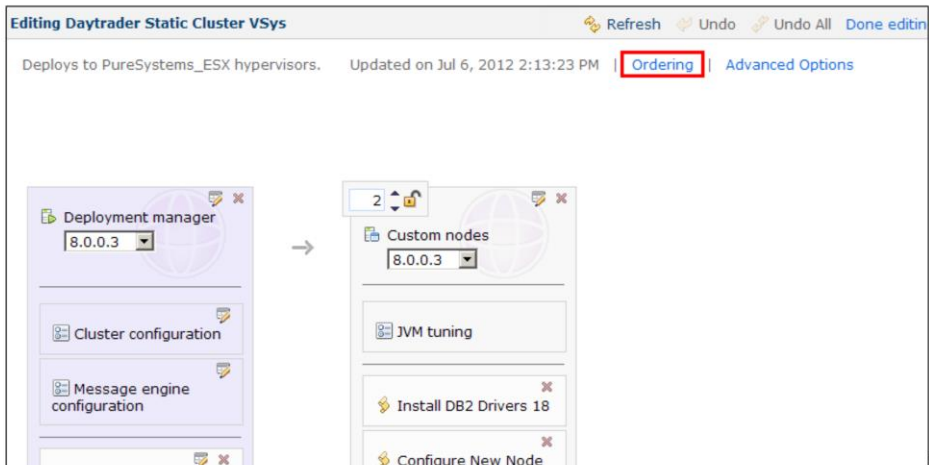
Some virtual image parts, like WebSphere Application Server, allow advanced options to further customize things like clustering or messaging for some of their parts. Enabling the advance options will automatically add some system scripts.

The example in this slide shows WebSphere Application Server Network Deployment topology cell with DB2, totaling four VMs: a deployment manager node managing two custom nodes and DB2, along with the system and user scripts.

Also shown are several features in the editor that allow you to customize the parts. You can specify the startup order, service levels, and system scripts from advanced options and user scripts.

Ordering – Startup order option for parts and scripts

- Allows user to specify and control the order of startup of different VMs (parts) and script packages during the deployment of the Virtual system pattern
 - For example, DB2 JDBC driver need to be installed and database need to be created before installing the application that accesses the database
- Click **Ordering** in pattern editor – this will bring up the order constraints of the parts and the script packages where the order can be changed



You can use the Pattern Editor to control to the order that different parts are deployed and started, and determine when the different script packages are run when the virtual system is deployed. Click the “Ordering” link to view and change the order.

By default, there are some startup ‘rules’ that govern a pattern. For instance, in a pattern that includes a “deployment manager” part, the “deployment manager” part will start before any custom nodes or HTTP servers. There are times when you might want to dictate the startup order of the parts beyond these default ‘rules’ though. You can do that in the pattern editor by specifying other constraints. You will see the default startup rules defined in your pattern as default constraints that you cannot alter or delete. You can add your own constraints beyond the default ones to dictate the order in which parts are started during deployment.

Order constraints

- Defined in terms of 'constraints' rather than straight 'order'
 - Default logical constraints exist – example: Cannot start custom node before deployment manager
- Order constraints defined separately for parts and scripts
- Move the scripts or parts up and down to change the order, as long as the default constraints are maintained

Part order constraints
 Custom nodes start after Deployment manager
 DB2 Enterprise starts after Custom nodes

Script order constraints
 Script Add IBM HTTP Server node in part Deployment manager starts after script Install DB2 Driver rg in part Custom nodes
 Script Install DayTrader Static Cluster Application rg in part Deployment manager starts after script create DayTrader DB2 Tables rg in part DB2 Enterprise
 Script Configure New Node rg in part Custom nodes starts after script Install DayTrader Static Cluster Application rg in part Deployment manager
 Script create DayTrader DB2 Tables rg in part DB2 Enterprise starts after script Install DB2 Driver rg in part Custom nodes

Default constraint – cannot be deleted

Selecting constraint highlights the parts/scripts

User constraint – can be deleted

1 Deployment manager
 2 Custom nodes
 3 DB2 Enterprise

1 Custom nodes
 2 Install DB2 Driver rg

2 Deployment manager
 Add IBM HTTP Server node

DB2 Enterprise

10 Creating virtual system patterns © 2013 IBM Corporation

This slide shows the startup order of the pattern parts and scripts that are defined in the pattern defined as constraints.

Order constraints are specified for the VM parts and scripts separately. The part order constraints dictate the order that the VMs in the pattern are deployed. Once all the parts are deployed, the script order constraints dictate when the scripts are run within the deployed VMs.

Also note that there are five parts but only four numbers to show the startup order as shown by the numbers with clocks next to them. This is because the second position in the startup order contains both custom node parts. This says that the two custom nodes parts can start in any order after the deployment manager starts. In fact they can be started asynchronously after the deployment manager starts.

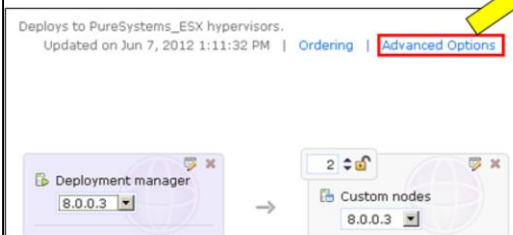
Default constraints cannot be deleted. You will note that the 'X' for the default constraint is not available. For example, custom nodes can only start after the deployment manager, as shown on the slide. The user constraints that you add to the pattern can be deleted. For example, DB2 starting after the custom node is not required, and so it is not a default constraint.

You can change the order to fit within these constraints.

Selecting the constraint will highlight the parts involved in the constraint. In the example on the page, selecting the first constraint in the script order constraint highlights the parts involved in the constraint.

Advanced options (customization)

- Advanced options is available for certain parts in your pattern to allow advanced customization on those parts
 - Currently available on WebSphere Application Server parts
- Allows you to configure advanced options for those parts
 - Clustering
 - Messaging
 - Session persistence
 - Global security



The following options are available for this pattern. The recommended options are common choices for topology patterns like this one.

- Define clusters**
 - Once enabled, the number of clusters and application servers can be configured on the deployment manager part.
- Enable messaging**
 - Additional message engine configuration will be available on the deployment manager part. Messaging engines point to WebSphere derby on the deployment manager. On the virtual system, start derby or configure the deployment manager to use another database.
 - Standard messaging engine configuration
 - Highly available messaging engine configuration
 - Scalable messaging engine configuration
 - Highly available and scalable messaging engine configuration
 - Enable MQ messaging (legacy feature)
 - MQ link configuration
 - MQ server configuration
- Enable session persistence**
 - Memory-memory implemented session persistence

OK Cancel

11

Creating virtual system patterns

© 2013 IBM Corporation

Based on the parts in your Virtual System pattern, certain advanced options are available. Advanced options provide a set of predefined configuration options of some common but more complex settings.

Not all parts have or need these options. Currently, the advanced options are available for the WebSphere Application Server part in your pattern. There might be other virtual image parts that are added to PureApplication System that will have their own advanced options.

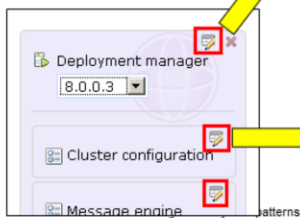
For WebSphere Application Server, the advanced options are for configuring clusters, messaging, and global security.

You can accomplish the same configuration by creating your own script packages. However, advanced options makes it easy for you to accomplish these configurations.

On enabling these options, certain script packages are automatically added to those parts. For example, setting “Define Cluster” in the advance option panel will add a “Define Cluster” script on the deployment manager.

Customizing pattern parts

- Based on the pattern part or script, you will need to provide some customized deployment properties, like passwords, memory size, and virtual CPUs
- Customize each virtual image part or script now or during deployment
- Customize script package environment variables
- Properties can optionally be locked, so they cannot change at deployment time



Properties for part Deployment manager (DMGRPart)

Name:	DMGRPart	
Virtual CPUs:	1	
Memory size (MB):	2048	
Reserve physical CPUs:	False	
Reserve physical memory:	False	
Cell name:	CloudBurstCell	
Node name:	CloudBurstNode	
Feature packs:	<input checked="" type="checkbox"/> none	

Lock property

OK Cancel

Parameters for script Cluster configuration

WEB_CLUSTER_PREFIX:	DayTraderCluster	
WEB_NUM_CLUSTERS:	1	
WEB_NUM_SERVERS_PER_NODE:	2	

OK Cancel

Corporation

Once you have assembled your high level topology, you can further configure your environment by clicking on the edit icon of each configurable virtual image part and script package to bring up a list of configurable options.

You can configure your environment during pattern creation or during pattern deployment. If you configure it at pattern creation, those values are available at deployment time as default values for the deployer to accept or modify for each pattern deployment.

You also have the option of locking down your configuration options. If for example, your company has a policy where all WebSphere Application Server cells are required to have a cell name "CompanyCell", enter "CompanyCell" into the "Cell name" field and click the lock icon located to the right of the field. The locked fields are shown during deployment but the deployer cannot change them.

Customizing pattern part – Common fields

- Each part in the pattern is given a part name by the system
 - Used by other parts (as environment variables) to get info about this part's VM – like IP address, host name, and so on.
- Each part has some common parameters
 - Number of Virtual CPUs – 1, 2, 4, 8
 - Image memory size
 - VM login “root” password
 - Enable VNC on the image
- Some parts have their own specific custom parameters that need to be configured

Each part in your pattern has its own set of required customization fields. There are some common fields that all parts need, like the number of virtual CPUs, VM memory size, VM OS root password, whether VNC needs to be enabled or not, and other values.

Specific parts can have their own fields. Like WebSphere Application Server needs WebSphere administration user ID and password; the deployment manager needs cell name and node name; and so on.

Part specific parameters in the virtual system patterns - Examples

- WebSphere Application Server - stand-alone server and deployment manager
 - For V7, enabling Feature packs (Batch, CEA, JPA, OSGI, SCA, SDO, XML)
 - These features are already included in WebSphere Application Server V8
 - Cell and node names
 - WebSphere Administrator user ID and password
- WebSphere Application Server – Custom node
 - Node name
- DB2
 - Passwords for db2inst1, db2fenc1, dausr1
 - Ability to create new or sample database
- WebSphere Message Broker
 - Broker name
 - Queue Manager information
 - Authorized users
 - Log information
 - HTTP and HTTPS information like address, port, and key store
- WebSphere MQ
 - Queue Manager information
 - Authorized users
 - Log information

This is only a partial list

This slide covers a partial list of specific parameters for certain virtual image parts. This is not an exhaustive list of parameters; neither does it contain parameters for all virtual image parts.

Virtual System pattern attributes

This section shows the attributes of the virtual system pattern.

Virtual system pattern attributes and options

- From this panel, you can:
 - Deploy, edit, clone, lock, and delete your pattern
 - View pattern topology
 - View the deployed systems using this pattern
 - View and modify access control – who can read or write your pattern
- Once locked, the pattern cannot be modified – you can clone the pattern and make changes to the clone

16

Creating virtual system patterns

© 2013 IBM Corporation

The functions of virtual system patterns shown on the top of the panel are deploy, edit, clone, lock, and delete. The “In the cloud now” section shows which virtual systems were created from this pattern and the “Access granted to” attribute shows who has permission to view, modify, and update this pattern.

There are two options that warrant a deeper look: the “Clone” and “Lock” pattern options.

Once you have created, tested, and verified your pattern you can lock down the pattern to prevent any further changes. Locking a pattern is a permanent operation; once locked it cannot be unlocked. If you later decide that changes are required, clone the pattern, make changes to the clone, test, and verify the changes. Once you are confident in your new pattern you can delete the original pattern.

Cloning a pattern makes an exact copy of the original. Typical use of the clone operation is to clone one of the predefined “best practices” patterns or any locked pattern, and use the result as a basis for your own custom patterns.

Section

Summary

And, in summary...

Summary

- Virtual system pattern creation
 - Drag-and-drop assembly
 - Specify the ordering of when Virtual system parts and script packages start during deployment
 - Configure pattern during deployment
- PureApplication System comes preloaded with industry “best practice” patterns

Creating a middleware topology is no longer an extensive manual process. With PureApplication System comes the pattern editor, which allows you to graphically edit your middleware topology.

Besides making it easy to create your own patterns, PureApplication System provides you with a set of industry “best practice” patterns that you can immediately use as-is.

Trademarks, disclaimer, and copyright information

IBM, the IBM logo, ibm.com, DB2, PureApplication, and WebSphere are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of other IBM trademarks is available on the web at "[Copyright and trademark information](http://www.ibm.com/legal/copytrade.shtml)" at <http://www.ibm.com/legal/copytrade.shtml>

Other company, product, or service names may be trademarks or service marks of others.

THE INFORMATION CONTAINED IN THIS PRESENTATION IS PROVIDED FOR INFORMATIONAL PURPOSES ONLY. WHILE EFFORTS WERE MADE TO VERIFY THE COMPLETENESS AND ACCURACY OF THE INFORMATION CONTAINED IN THIS PRESENTATION, IT IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. IN ADDITION, THIS INFORMATION IS BASED ON IBM'S CURRENT PRODUCT PLANS AND STRATEGY, WHICH ARE SUBJECT TO CHANGE BY IBM WITHOUT NOTICE. IBM SHALL NOT BE RESPONSIBLE FOR ANY DAMAGES ARISING OUT OF THE USE OF, OR OTHERWISE RELATED TO, THIS PRESENTATION OR ANY OTHER DOCUMENTATION. NOTHING CONTAINED IN THIS PRESENTATION IS INTENDED TO, NOR SHALL HAVE THE EFFECT OF, CREATING ANY WARRANTIES OR REPRESENTATIONS FROM IBM (OR ITS SUPPLIERS OR LICENSORS), OR ALTERING THE TERMS AND CONDITIONS OF ANY AGREEMENT OR LICENSE GOVERNING THE USE OF IBM PRODUCTS OR SOFTWARE.

© Copyright International Business Machines Corporation 2013. All rights reserved.