

Using IBM Urban Code deploy to define queues and channels onto target IBM MQ For z/OS Environments

Please note

- IBM's statements regarding its plans, directions, and intent are subject to change or withdrawal without notice at IBM's sole discretion.
- Information regarding potential future products is intended to outline our general product direction and it should not be relied on in making a purchasing decision.
- The information mentioned regarding potential future products is not a commitment, promise, or legal obligation to deliver any material, code or functionality. Information about potential future products may not be incorporated into any contract.
- The development, release, and timing of any future features or functionality described for our products remains at our sole discretion.
- Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput or performance that any user will experience will vary depending upon many factors, including considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve results similar to those stated here.

Agenda

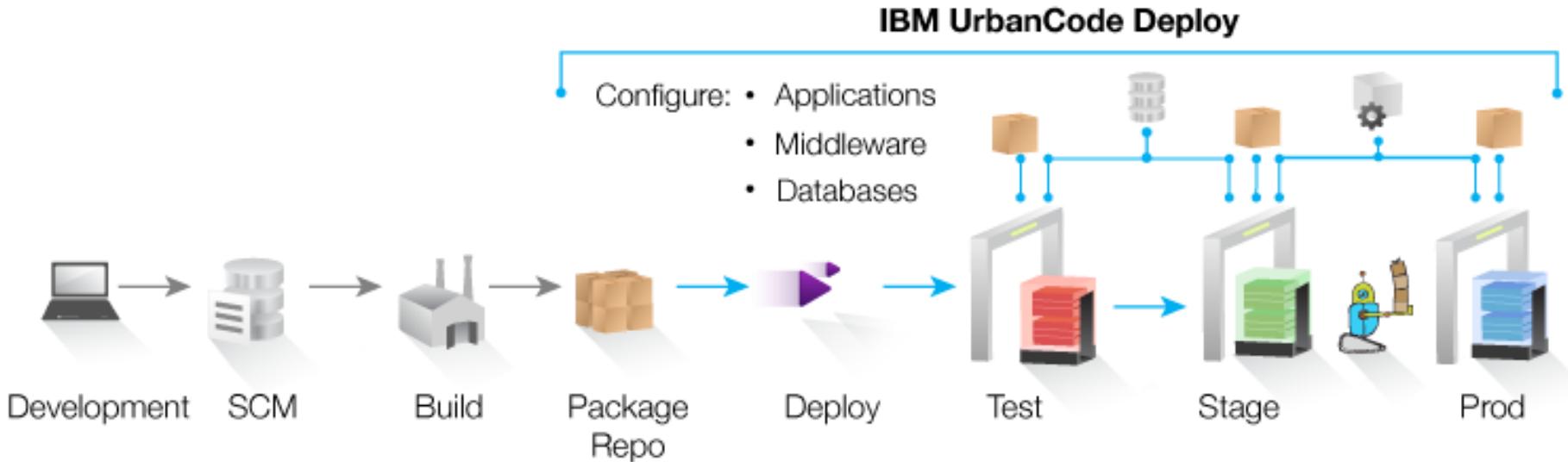
- What is IBM Urban Code Deploy ?
- MQ Provisioning Technology
- MQ Plugins for UCD
- Distribution of the MQ for z/OS plugin for UCD
- What do you need to run the MQ for z/OS plugin for UCD ?
 - UCD Server, UCD Agent, UCD Artifacts
- Supported Software Levels
- Does the MQ for z/OS plugin run on distributed platforms ?
- MQ Resource Deployments
- Example Bank Payments Application
- Sample Triplet Files
- Source Code
- Trace
- Further Information
- Q&A

What is IBM Urban Code Deploy ?

What is IBM Urban Code Deploy ?

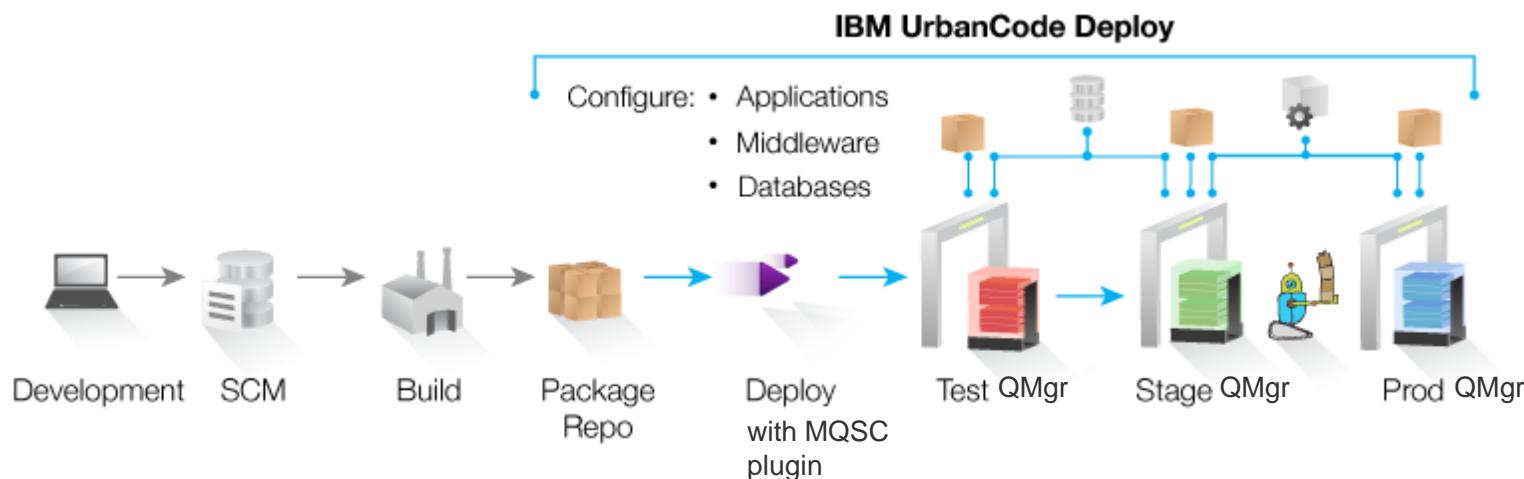
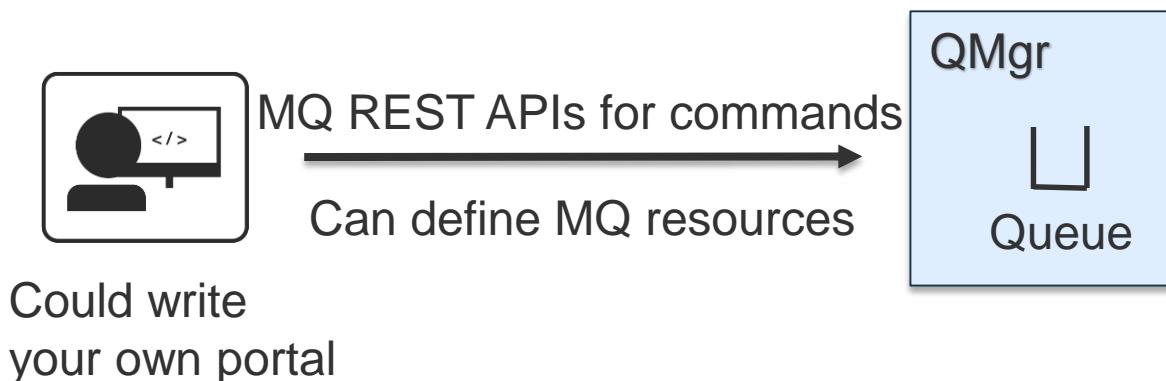
- IBM® UrbanCode™ Deploy **orchestrates and automates application deployments, middleware configurations, and database changes to on premise or cloud-based development, test, and production environments.**
- Teams can deploy as often as needed—**on demand or on schedule, with security-rich, self-service release management.**
- Whether you require on-premise or as-a-service, IBM UrbanCode Deploy helps you **accelerate your time to market, drive down deployment costs, reduce risks, and achieve continuous delivery.**
- For further information please see: <http://www-03.ibm.com/software/products/en/ucdep>

Deployment using IBM Urban Code Deploy



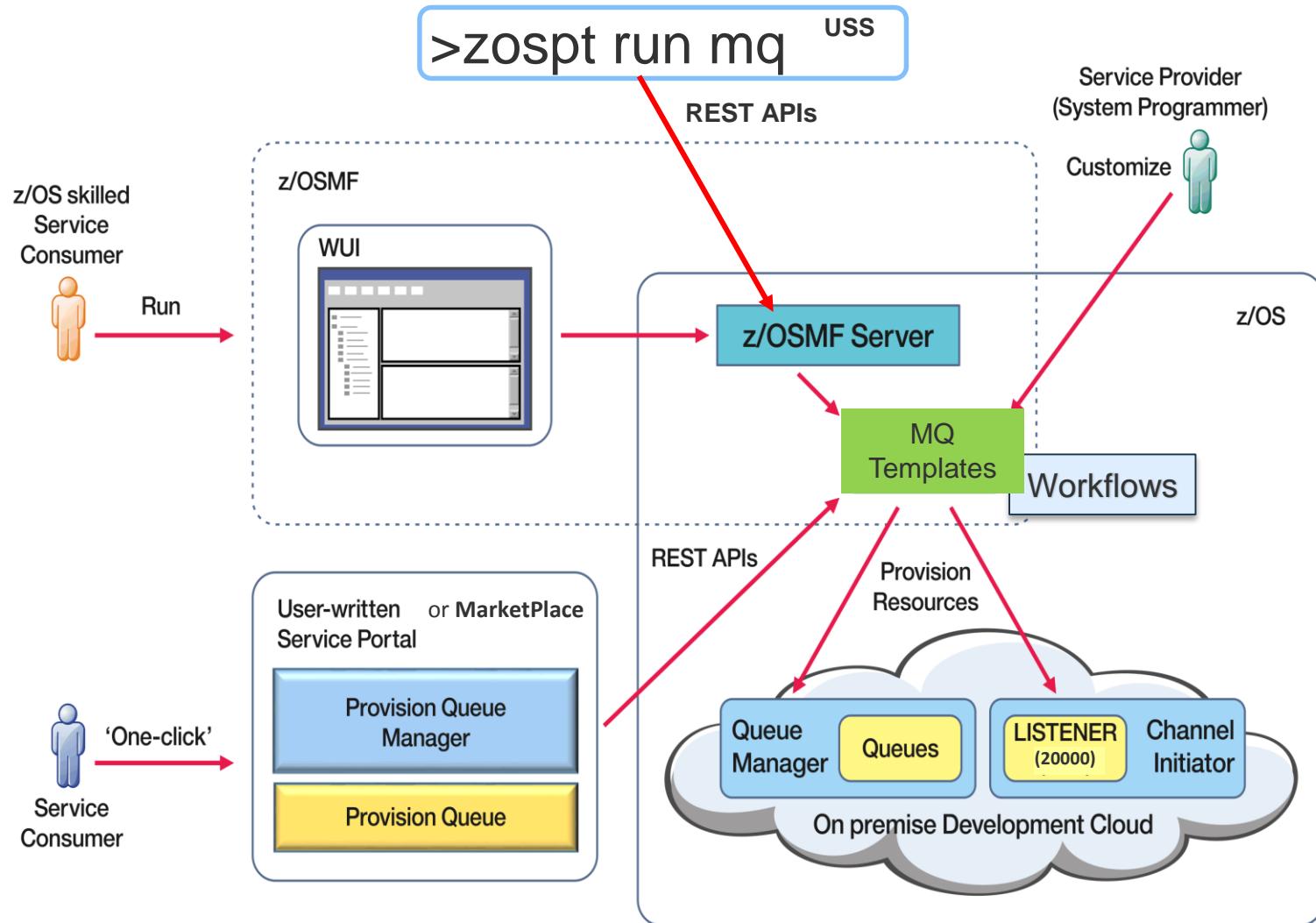
MQ Provisioning Technology

MQ Provisioning Technology



- With UCD, Queues and Channels can be defined in a file system or SCM, and deployed to target environments
- UCD has capability to automatically change the characteristics of resources as they are deployed from one environment to another

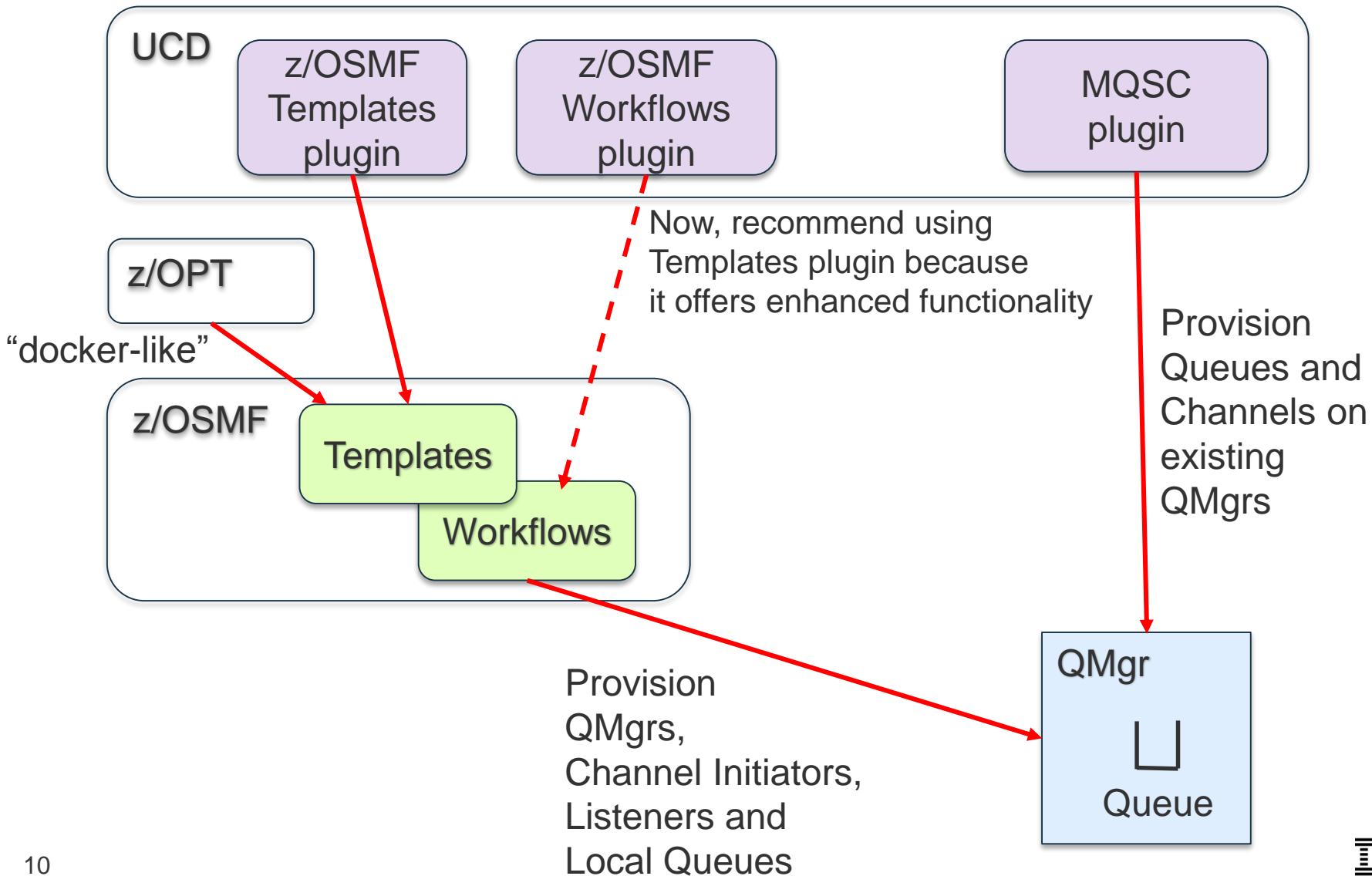
MQ Provisioning Technology



- Can provision Queue Managers, Channel Initiators, Listeners and Queues
- No support for staged deployment to environments but, workflows can be designed to introduce environments

MQ Provisioning Technology

- Some customers would like to use UCD as the overall orchestration engine



MQ Provisioning Technology

MQ Deploy Support	Distributed Platforms	z/OS
Queue Manager	IBM MQ (distributed) plugin - dedicated plugin steps to create/start/stop/delete Queue Managers	UCD JCL and MVS Command steps
	Docker containers. New technology is evolving for MQ in the cloud (i.e. services in IBM Cloud and other Clouds).	z/OSMF workflows z/OS Provisioning Toolkit z/OSMF plugin for UCD
Queue, Channel	IBM MQ (distributed) plugin - dedicated plugin steps to create/delete MQ queue, channel and other resources	IBM MQ for zOS Generate MQSC Commands plugin - Step to generate MQSC commands from meta file artifacts and submit JCL (CSQUTIL)
	IBM MQ REST API	IBM MQ REST API
	Docker containers. New technology is evolving for MQ in the cloud (i.e. services in IBM Cloud and other Clouds).	z/OSMF workflows (some) z/OS Provisioning Toolkit (some) z/OSMF plugin for UCD
		UCD batch job submit step to run CSQUTIL job to issue MQSC commands.
General commands	IBM MQ (distributed) plugin - dedicated plugin step to run MQSC commands	UCD batch job submit step to run CSQUTIL job to issue MQSC commands.
	Docker containers. New technology is evolving for MQ in the cloud (i.e. services in IBM Cloud and other Clouds).	

MQ Plugins for UCD

MQ Plugins for Urban Code Deploy

- **There are two plugins available:**
 - **IBM-MQ (distributed) plugin**
 - Issues MQ control commands and MQSC commands via `runmqsc` to create MQ for distributed Queue Managers and define Queues, Channels, etc. on them
 - **IBM MQ for zOS Generate MQSC Commands plugin**
 - Defines a JSON format for representing resources in a triplet of files
 - Processes the files to generate MQSC commands to define Queues and Channels on MQ for z/OS Queue Managers
 - Source code is available for download
- **Both plugins are fully supported by IBM Service Teams**

IBM MQ (distributed) plugin

<https://developer.ibm.com/urancode/plugin/websphere-mq/>

IBM-MQ

8

Download

IBM® MQ is robust messaging middleware that simplifies and accelerates the integration of diverse applications and business data across multiple platforms. IBM MQ facilitates the assured, secure and reliable exchange of information between applications, systems, services and file by sending and receiving message data via messaging queues, thereby simplifying the creation and maintenance of business applications. It delivers Universal Messaging with a broad set of offerings to meet enterprise-wide messaging needs, as well as connectivity for the internet of things and mobile devices. With this plugin, you can directly interact with WebSphere MQ with dozens of out of the box steps.

Version 8

Release Notes: (Released November 14, 2016)

- Single quotes will be escaped as necessary. Previously escaped single quotes will be untouched.
- GroovyRuntimeException: Stream Closed error is resolved.

Other Versions:

8

4

2

1.423574



Product	IBM UrbanCode Deploy
Type	Plugin
Compatibility	IBM UrbanCode 6.0 and up
Created By	IBM UrbanCode
Need help?	See Documentation View Questions Ask a Question

[Download Past Plug-in Versions](#)

IBM MQ (distributed) plugin

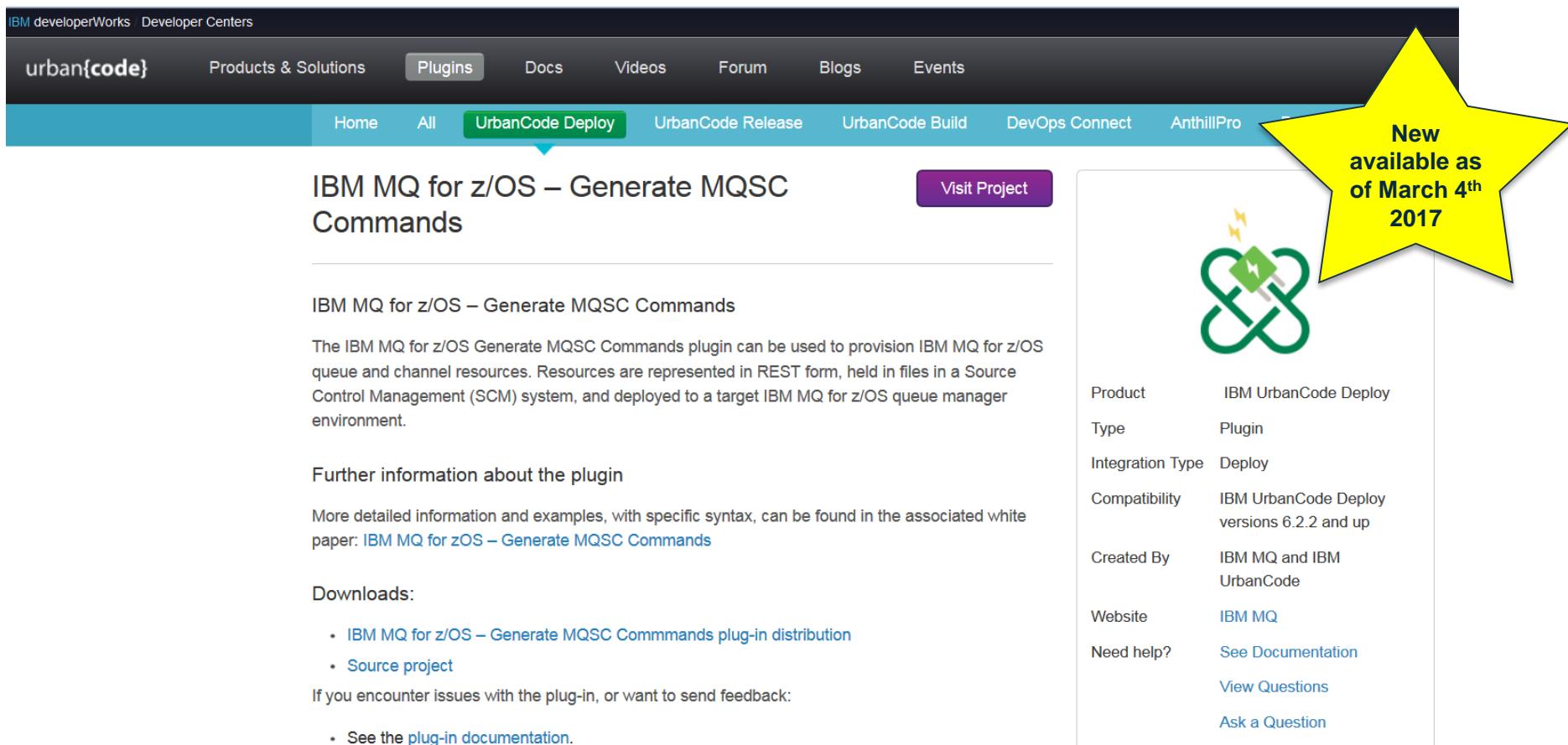
- WebSphereMQ / Alter Client-Connection Channel
- WebSphereMQ / Alter Queue
- WebSphereMQ / Alter Receiver Channel
- WebSphereMQ / Alter Sender Channel
- WebSphereMQ / Alter Server-Connection Channel
- WebSphereMQ / Alter Topic
- WebSphereMQ / Configure Channel SSL
- WebSphereMQ / Configure Queue Manager SSL
- WebSphereMQ / Create Queue Manager
- WebSphereMQ / Define Client-Connection Channel
- WebSphereMQ / Define Namelist
- WebSphereMQ / Define Queue
- WebSphereMQ / Define Queue Alias
- WebSphereMQ / Define Receive Channel
- WebSphereMQ / Define Sender Channel
- WebSphereMQ / Define Topic
- WebSphereMQ / Define Topic Subscription

- WebSphereMQ / Delete Certificate
- WebSphereMQ / Delete Channel
- WebSphereMQ / Delete Namelist
- WebSphereMQ / Delete Queue
- WebSphereMQ / Delete Queue Alias
- WebSphereMQ / Delete Queue Manager
- WebSphereMQ / Delete Topic
- WebSphereMQ / Delete Topic Subscription
- WebSphereMQ / Enable Queue Triggering
- WebSphereMQ / Import Certificate
- WebSphereMQ / Manage Queue Depth
- WebSphereMQ / Receive Certificate
- WebSphereMQ / Run MQ Batch Script
- WebSphereMQ / Run MQ Script

- WebSphereMQ / Set Channel Monitoring Level
- WebSphereMQ / Set Max Message Size On Queue
- WebSphereMQ / Set Permissions For Object
- WebSphereMQ / Set Permissions For Queue
- WebSphereMQ / Set Permissions For Topic
- WebSphereMQ / Set Queue Usage Type
- WebSphereMQ / Start Channel
- WebSphereMQ / Start Queue Manager
- WebSphereMQ / Stop Channel
- WebSphereMQ / Stop Queue Manager

IBM MQ for z/OS Generate MQSC Commands plugin

<https://developer.ibm.com/urbancode/plugin/ibm-mq-for-zos/>



IBM developerWorks / Developer Centers

urban{code} Products & Solutions Plugins Docs Videos Forum Blogs Events

Home All UrbanCode Deploy UrbanCode Release UrbanCode Build DevOps Connect AnthillPro

IBM MQ for z/OS – Generate MQSC Commands

[Visit Project](#)

IBM MQ for z/OS – Generate MQSC Commands

The IBM MQ for z/OS Generate MQSC Commands plugin can be used to provision IBM MQ for z/OS queue and channel resources. Resources are represented in REST form, held in files in a Source Control Management (SCM) system, and deployed to a target IBM MQ for z/OS queue manager environment.

Further information about the plugin

More detailed information and examples, with specific syntax, can be found in the associated white paper: [IBM MQ for zOS – Generate MQSC Commands](#)

Downloads:

- [IBM MQ for z/OS – Generate MQSC Commands plug-in distribution](#)
- [Source project](#)

If you encounter issues with the plug-in, or want to send feedback:

- See the [plug-in documentation](#).
- Ask IBM MQ for z/OS – Generate MQSC Commands plug-in specific questions on the [GitHub Issues](#) page.
- For general plug-in issues, submit questions to the [dW Answers](#) page.
Use the following tags: urbancode | ucdev | plugins

Product	IBM UrbanCode Deploy
Type	Plugin
Integration Type	Deploy
Compatibility	IBM UrbanCode Deploy versions 6.2.2 and up
Created By	IBM MQ and IBM UrbanCode
Website	IBM MQ
Need help?	See Documentation View Questions Ask a Question

Distribution of MQ for z/OS plugin for UCD

MQ for z/OS plugin distribution (on github.com)

The screenshot shows the GitHub repository page for `IBM-UrbanCode/IBM-MQ-zOS-UCD`. The top navigation bar includes links for Features, Business, Explore, Marketplace, Pricing, and Sign in or Sign up. The repository header shows 80% completion, a search bar, and options to Watch (8), Star (1), Fork (1). The main navigation tabs are Code, Issues (0), Pull requests (0), Projects (0), and Insights. The 'Releases' tab is selected. A green box highlights the 'Latest release' button. Below it, there are two download links: 'MQ-zOS-UCD-v2.4c25d78.zip' (3.07 MB) and 'Source code (zip)'. The 'Source code (tar.gz)' link is also present.

Latest release
MQ-zOS-UCD-v2.4c25d78
nhmathis released this on 15 Mar · 2 commits to master since this release

Release Notes Version 2:
Resolved MQSCUtil missing class exception.

Downloads

- MQ-zOS-UCD-v2.4c25d78.zip
- Source code (zip)
- Source code (tar.gz)

The UCD Development Team is:

- Gradually moving all plugins into github
- Making plugin source code available



Plugin source (on github.com)

GitHub, Inc. (US) | https://github.com/IBM-UrbanCode/IBM-MQ-zOS-UCD?cm_mc_uid=57911109999414937207513&cm_mc_sid_50200000= 80% Search

Features Business Explore Marketplace Pricing This repository Search Sign in or Sign up

IBM-UrbanCode / IBM-MQ-zOS-UCD Watch 8 Star 1 Fork 1

Code Issues 0 Pull requests 0 Projects 0 Insights ▾

IBM MQ z/OS plugin for IBM UrbanCode Deploy used to Generate MQSC Commands. <https://developer.ibm.com/urbancode/p...>

urbancode urbancode-deploy ibm-mq zos

24 commits 1 branch 2 releases 3 contributors

Branch: master New pull request Find file Clone or download ▾

gurugg Update examples Latest commit ac05fee on 13 Apr

doc	Update examples	2 months ago
license	Standarized to other OSS Plugins	4 months ago
plugin	Spelling	3 months ago
src	Rename mqscutil.groovy to MQSCUtil.groovy	3 months ago
.gitignore	Updated Travis Support	4 months ago
.travis.yml	Updated Travis Support	4 months ago
README.md	Updated zip location	3 months ago
build.xml	Explode groovy-plugin-utils into the classes folder	4 months ago
ivy.xml	Explode groovy-plugin-utils into the classes folder	4 months ago

README.md

IBM UrbanCode Deploy MQ z/OS - Generate MQSC Commands Plugin

What do you need to run the
MQ for z/OS plugin for UCD ?

What do you need to run the MQ for z/OS UCD plugin ?

- A Source Control Management System (e.g. RTC) or, can also read straight from a file system
- A UCD server (runs as a Liberty Was Profile server)
- A UCD agent (preferably running on a z/OS system)
- Artifacts need to be created in UCD (see white paper)
- MQ for z/OS Queue Manager
 - You can use an existing Queue Manager
 - Or you can provision a new Queue Manager using z/OSMF templates and workflows.
 - **UCD Development have recently provided a new plugin for this (see next slide)**

UCD Artifacts

- Artifacts need to be created in UCD
 - Details of the artifacts you need to create are documented in the white paper. See: https://github.com/IBM-UrbanCode/IBM-MQ-zOS-UCD/tree/master/doc/white_paper

Please do take a look at the white paper !

New UCD Plugin – discover/create/manage z/OSMF workflows and templates

https://developer.ibm.com/urbancode/plugin/zos-management-facility/

IBM developerWorks Developer Centers

urban{code} Products & Solutions Plugins Docs Videos Forum Blogs Events

Home All UrbanCode Deploy UrbanCode Release UrbanCode Build DevOps Connect AnthillPro Development Community

zOS Management Facility

6

The z/OS Management Facility (zOSMF) plug-in provides functions to discover software instances provisioned by z/OSMF Cloud Provisioning or z/OS Provisioning Toolkit.

The zOS Management Facility plug-in includes steps to create and manage z/OSMF workflows and software service templates.

Platform Support:

- z/OSMF Cloud Provisioning is available on z/OS V2R2 (PTF UI42847, UI46543), V2R1 (PTF UI43814) and later
- This plug-in requires UrbanCode Deploy 6.2.0 or later. Specific resource icons for z/OS software instances are used in UrbanCode Deploy 6.2.6 and later

Mainframe

Version 6

Release Notes: (Released September 19, 2017)

Added zOSMF discovery feature. The plug-in is now able to discover software instances provisioned by zOSMF Cloud Provisioning or zOS Provisioning Toolkit. A set of resources representing the software instances are automatically created as the result of the discovery. Added a new step to Run Published Software Service Template.

Download



Product	IBM UrbanCode Deploy
Type	Plugin
Compatibility	UrbanCode Deploy 6.2.0 and later
Created By	IBM UrbanCode
Need help?	See Documentation View Questions Ask a Question

Download Past Plug-in Versions

Released
in
September
2017



Supported Software Levels

What software levels does the MQ for z/OS UCD plugin support ?

- IBM z/OS V2.1 (or above)
- IBM MQ for z/OS V8 (or above)
 - As the plugin generates MQSC commands from the JSON form resource representations, it is important to ensure that the generated MQSC is supported by the version of IBM MQ for z/OS in use
 - The list of supported MQSC attributes for queue and/or channel resources can be found in the IBM MQ for z/OS Knowledge Center specific to the version of IBM MQ for z/OS in use
 - Attempts to use unsupported MQSC attributes will result in a failure of the MQSC command on the target MQ for z/OS queue manager environment
- IBM Urban Code Deploy version 6.2.2 (or later)
- Groovy version 1.8.8 (or later)

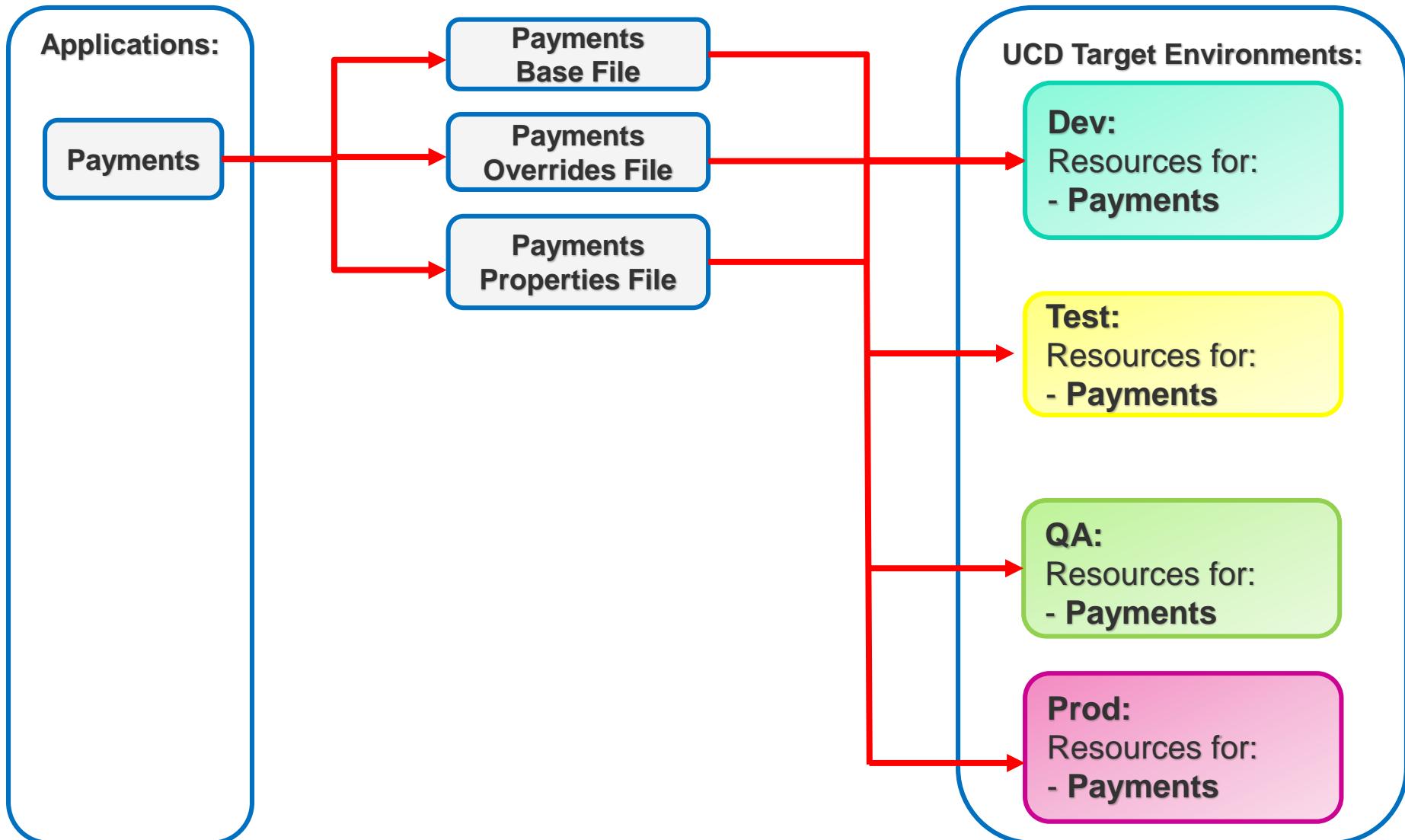
Does the MQ for z/OS Plugin run on
distributed platforms ?

Does the MQ for z/OS plugin run on distributed MQ ?

- NO, would require further development to run on distributed
- Would need to:
 - Drive runmqsc to issue commands on distributed
 - Address platform specific attributes. e.g. STGCLASS, CMDSCOPE, etc.
- In future, the plugin could be modified to:
 - Issue MQ REST APIs instead of MQSC via batch job
 - This would simplify the plugin drastically
 - Also, if the MQ REST API (which is still evolving) is smart enough to deal with platform specific attributes, we would get this function for free !

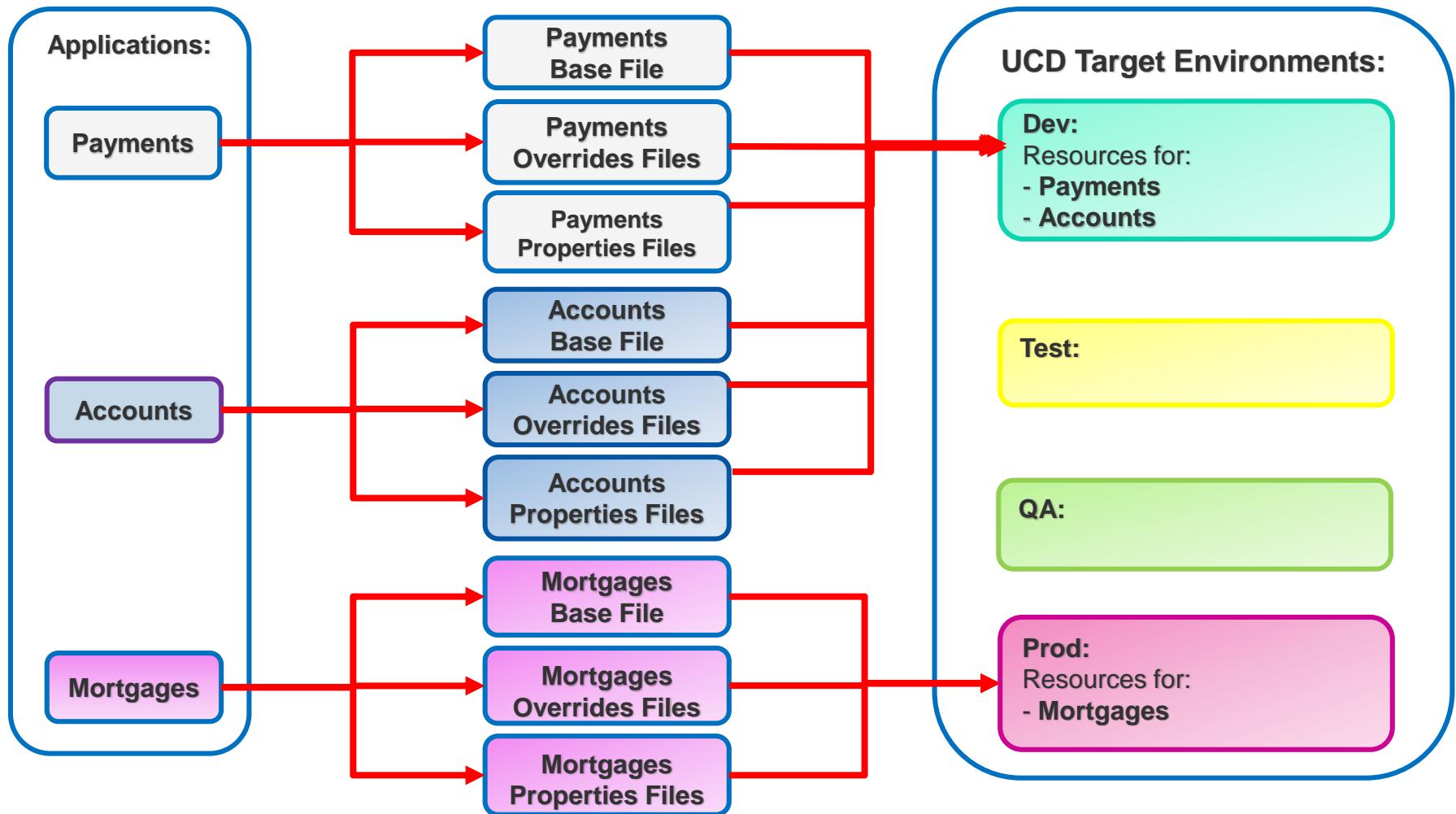
MQ resource deployments

MQ Applications and resource deployments

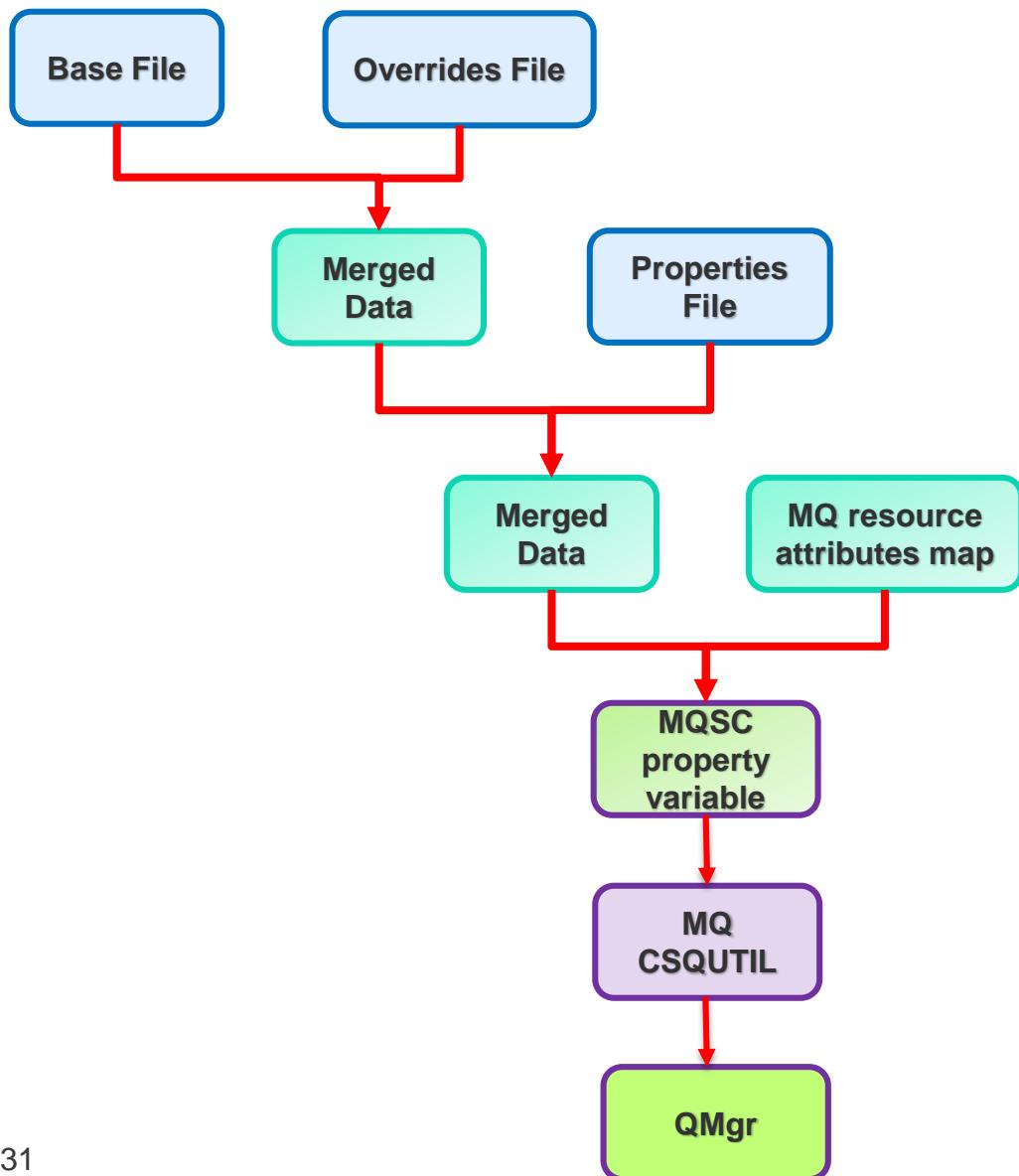


Note: 1) Developers focus on the application and on the JSON in the triplet of files
2) Developers do not need deep knowledge of MQ

MQ Applications and resource deployments



Processing the triplet of files



- Apply requested overrides to Base File
- Store merged data
- Apply properties to merged data
- Use map to Convert JSON to MQSC form
- Store MQSC form into a variable
- Use variable with the MQ CSQUTIL job
- Define resources to the MQ for z/OS Queue Manager

Deployment of resources to target environments

Base File:

```
{  
  "resource": "queue",  
  "type": "local",  
  "name": "@Q@",  
  "command": "DEFINE",  
  "maximumDepth":  
    2000  
}
```

Overrides File:

Deployment Targets:

Dev:

Test:

```
{"maximumDepth": 4000}
```

QA:

```
{"maximumDepth": 6000}
```

Prod:

```
{"maximumDepth": 8000}
```

Properties File:

```
{Q=PAYQ}
```

UCD Target Environments:

Dev:
DEFINE QLOCAL(PAYQ)
MAXDEPTH(2000)

Test:
DEFINE QLOCAL(PAYQ)
MAXDEPTH (4000)

QA:
DEFINE QLOCAL(PAYQ)
MAXDEPTH (6000)

Prod:
DEFINE QLOCAL(PAYQ)
MAXDEPTH (8000)

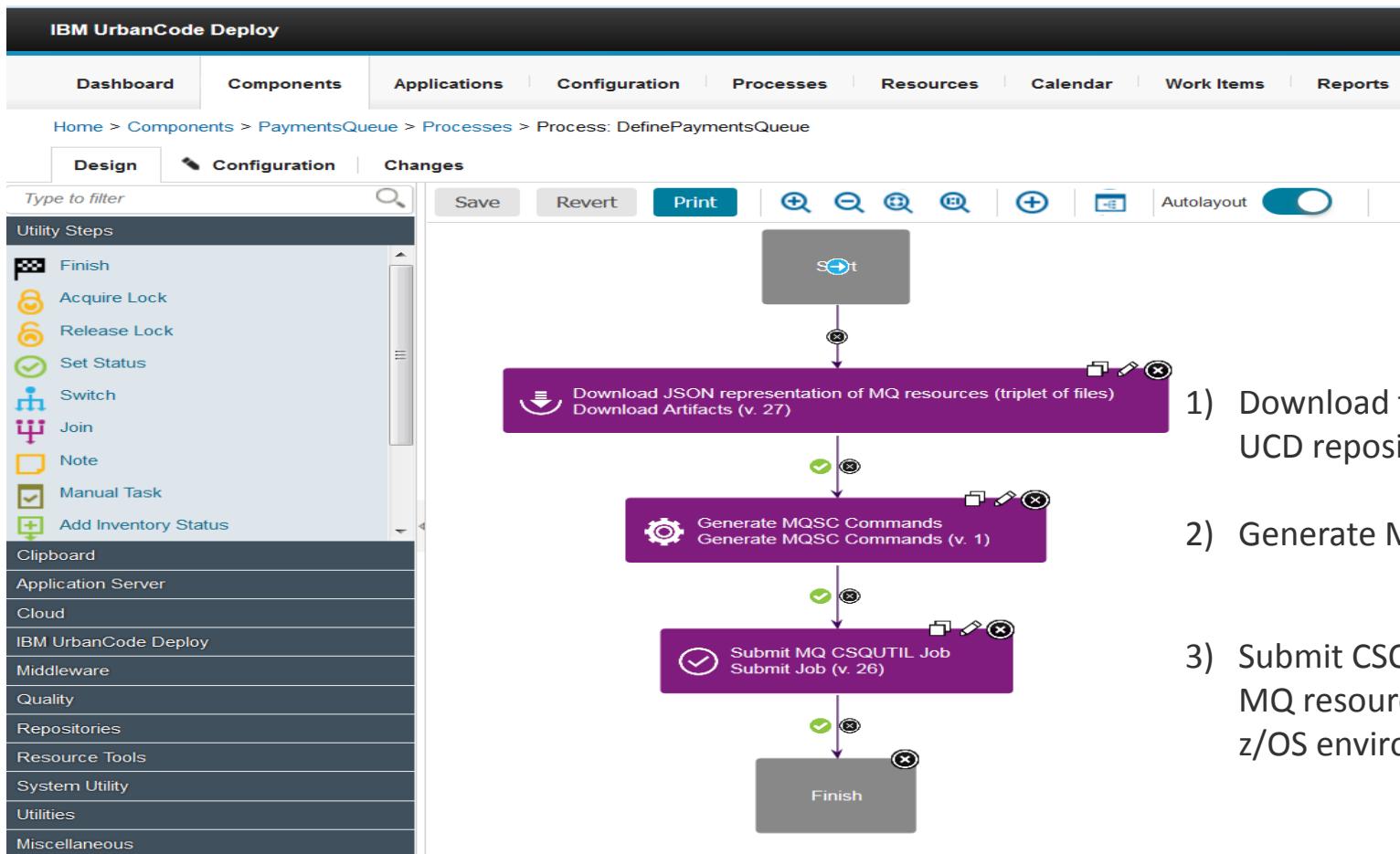
Deployment of resources to target environments

- The IBM MQ for z/OS Generate MQSC Commands plugin for IBM Urban Code Deploy (UCD) allows users to deploy MQ queue and channel resources, represented in JSON form and held in a triplet of files in a Source Control Management (SCM) system, onto a target MQ for z/OS Queue Manager environment defined in UCD. The files consist of:
 - **base** MQ resource definitions,
 - **override** values per target deployment environment,
 - **properties** values.
- The example, which uses a slightly simplified syntax for illustrative purposes, demonstrates the deployment of a local queue.

Notice how the maximum depth of local queue PAYQ varies depending on the target deployment environment. The local queue name and other queue attributes can, in a similar way, be varied based on the target environment. Detailed examples, with specific syntax, can be found in the white paper that accompanies the plugin.

- For full details about the plugin and the white paper, see: <https://developer.ibm.com/urbancode/plugin/ibm-mq-for-zos/>
- Note that the JSON format used to define Queues and Channels is based on the MQ REST APIs for defining Queues and Channels. However, at the time of publishing the plugin, the REST APIs for Channels was still in development so the intent is to factor any changes in the Channel REST APIs into the plugin at a later date.

Simple UCD process to define MQ for z/OS resources



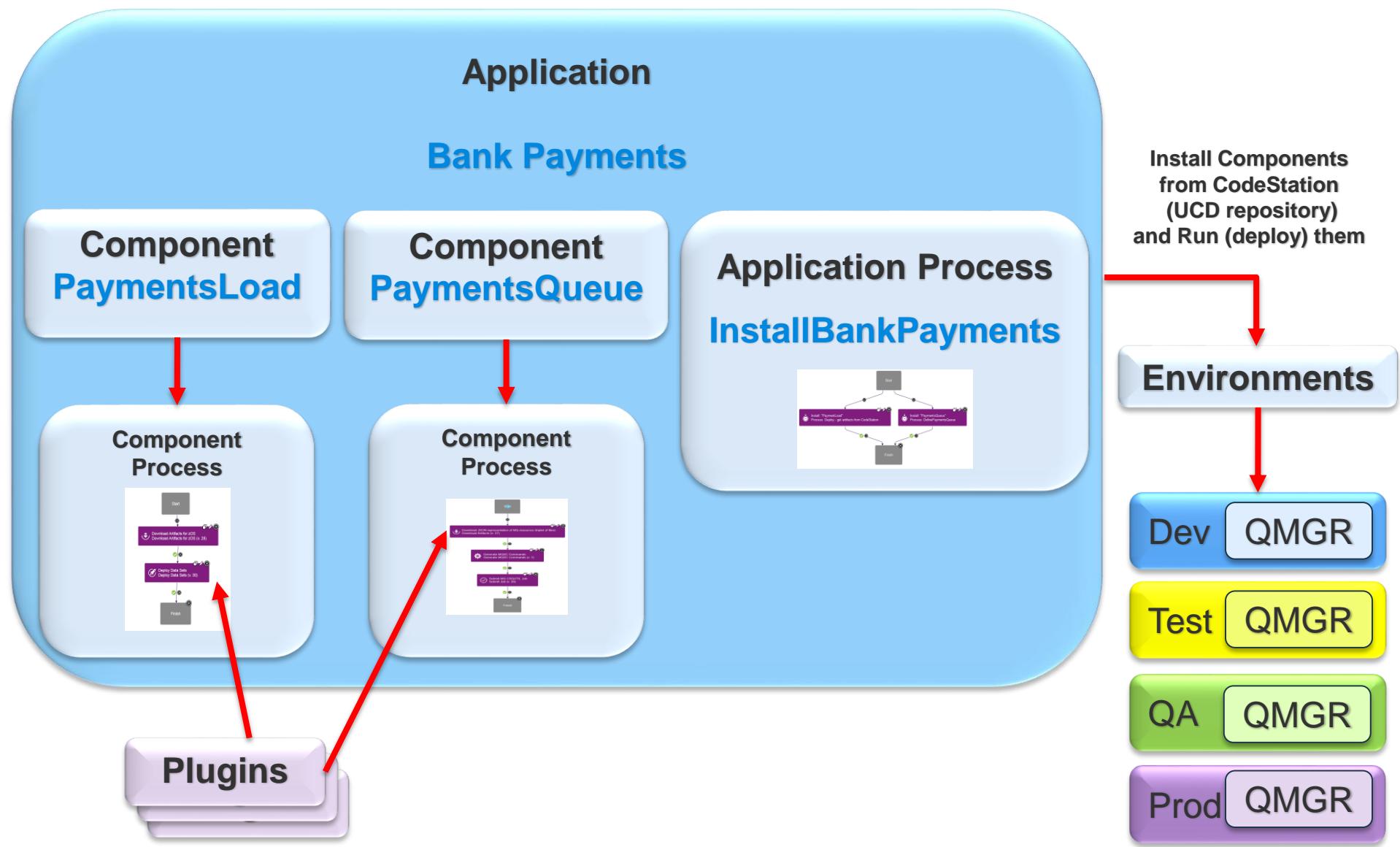
- 1) Download files from UCD repository
- 2) Generate MQSC commands
- 3) Submit CSQUTIL job to define MQ resources to target MQ for z/OS environment

Simple UCD process to define MQ for z/OS resources

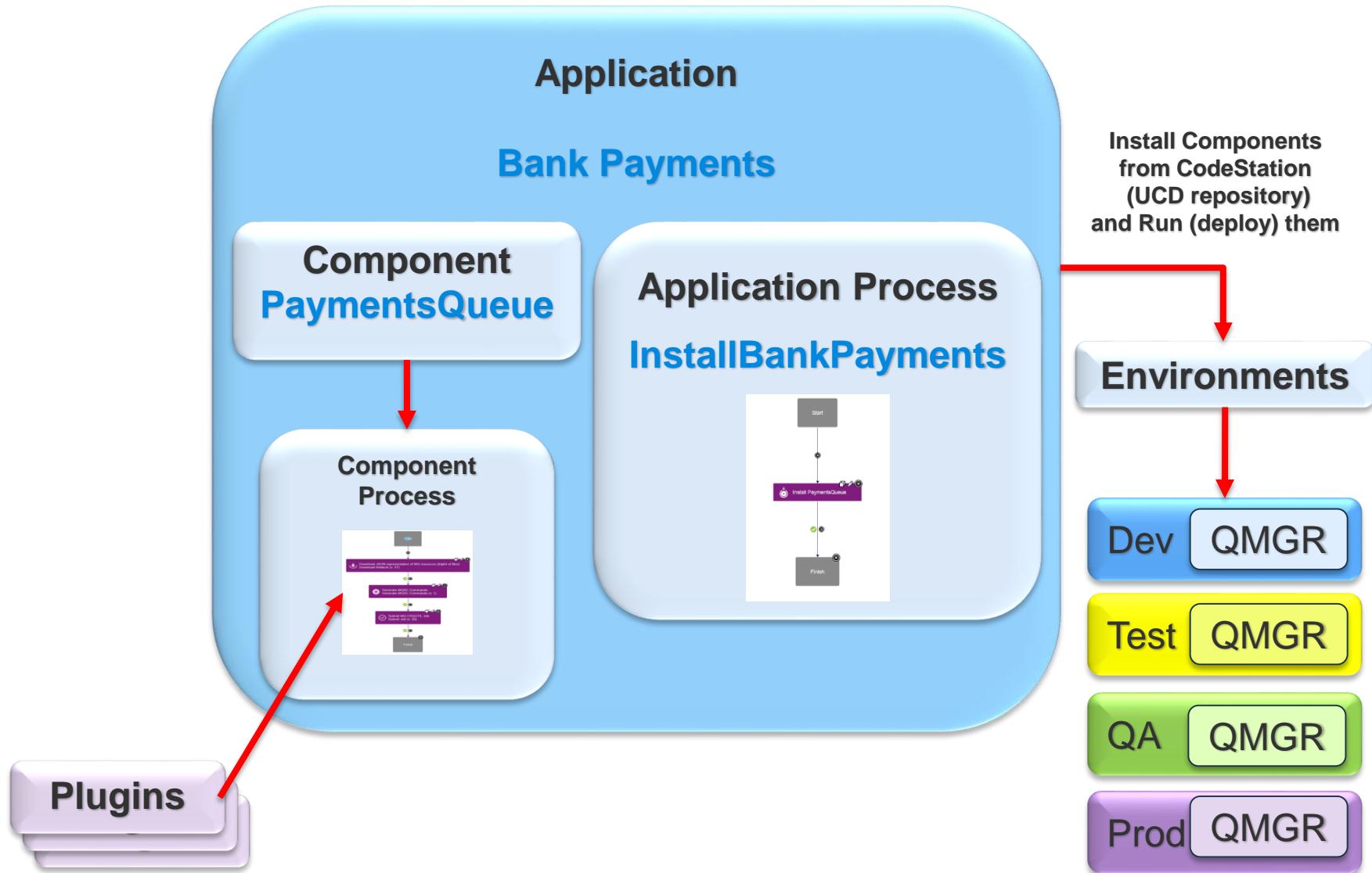
- The plugin is designed to be defined as a step in an IBM UCD process so that when the process is deployed against a target IBM MQ for z/OS Queue Manager environment, resources are automatically defined with the attribute values specified for the target environment.
- In the example, the first step downloads the triplet of files, that had previously been imported from the SCM system into the UCD repository, the second step converts the JSON form into the MQSC form, and the third step submits the IBM MQ CSQUTIL batch job to run the MQSC commands against the target IBM MQ for z/OS Queue Manager environment to define the MQ resources.

Example Bank Payments Application

UCD Artefacts for Bank Payments Application



UCD Artefacts for Bank Payments Application (as per artifacts documented in the WhitePaper)



Bank Payments Application

File Edit View History Bookmarks Tools Help

IBM Hursley z/OS Provisioning x | Tools As A Service x | TaaS Status x | CITM zOS Information - Vir... x | IBM-MQ-zOS-UCD/mqRes... x | IBM UrbanCode Deploy: A... x +

groovy script

IBM UrbanCode Deploy

M. Raja

Dashboard Components Applications Configuration Processes Resources Calendar Work Items Reports Settings

Home > Applications

Applications Templates

Create Application Import Applications Actions... Flat list

Name	Template	Description	Created	By
AutorollbackApp			5/5/2015, 10:50 AM	admin (use only when necessary) (admin)
Bank Payments	Actions...	Bank Payments Application	10/9/2016, 8:13 PM	M. Raja (mayur_raja@uk.ibm.com)
BreadboxApp			12/13/2015, 7:52 AM	admin (use only when necessary) (admin)
ChangemanApp			5/5/2015, 6:59 AM	admin (use only when necessary) (admin)
CICSWebServiceApp			10/8/2016, 7:19 AM	ZHANG HONG CHEN (chenzhh@cn.ibm.com)
Combined Application	CombinedDemo x		10/5/2015, 2:18 PM	admin (use only when necessary) (admin)
Corporate Banking	mq x		9/14/2016, 9:11 PM	Michael S. Samano (msamano@us.ibm.com)
DST APP1			4/28/2016, 10:02 AM	ZHANG HONG CHEN (chenzhh@cn.ibm.com)
ForceApp		MVSApplication	1/16/2016, 12:43 PM	admin (use only when necessary) (admin)
Hello Bluemix			8/28/2014, 9:16 PM	admin (use only when necessary) (admin)

22 records - Refresh Print

1 / 3

Rows 10

Bank Payments Application - Environments

Screenshot of the IBM UrbanCode Deploy application interface showing the environments for the Bank Payments application.

The browser tabs show multiple IBM-related sites: IBM Hursley z/OS Provisioning, Tools As A Service, TaaS Status, CITM zOS Information - Vir..., IBM-MQ-zOS-UCD/mqRes..., and IBM UrbanCode Deploy: B... . The search bar contains "groovy script".

The main navigation bar includes links for Dashboard, Components, Applications, Configuration, Processes, Resources, Calendar, Work Items, Reports, and Settings.

The breadcrumb navigation shows Home > Applications > Bank Payments.

The application title is Application: Bank Payments (show details).

The current view is the Environments tab, with other tabs available: History, Configuration, Components, Blueprints, Snapshots, Processes, Calendar, and Changes.

A prominent button at the top left says "Create Environment".

A message at the top right says "Drag environments by their names to re-order them. 6 Environments".

Search filters include "Search by Name" and "Search by Blueprint". Buttons for "Expand All" and "Collapse All" are also present.

The environment list displays six entries:

Environment Name	Snapshot Status	Compliance Status
Dev	Snapshot: None	Compliance: 3 / 12
Test	Snapshot: None	Compliance: 1 / 1
TEST ZOS COMPONENT	Snapshot: None	Compliance: 0 / 0
QA	Snapshot: None	Compliance: 0 / 0
Prod	Snapshot: None	Compliance: 0 / 0
M900 ON MVS043	Snapshot: None	Compliance: 1 / 2

Bank Payments Application – InstallBankPayments process

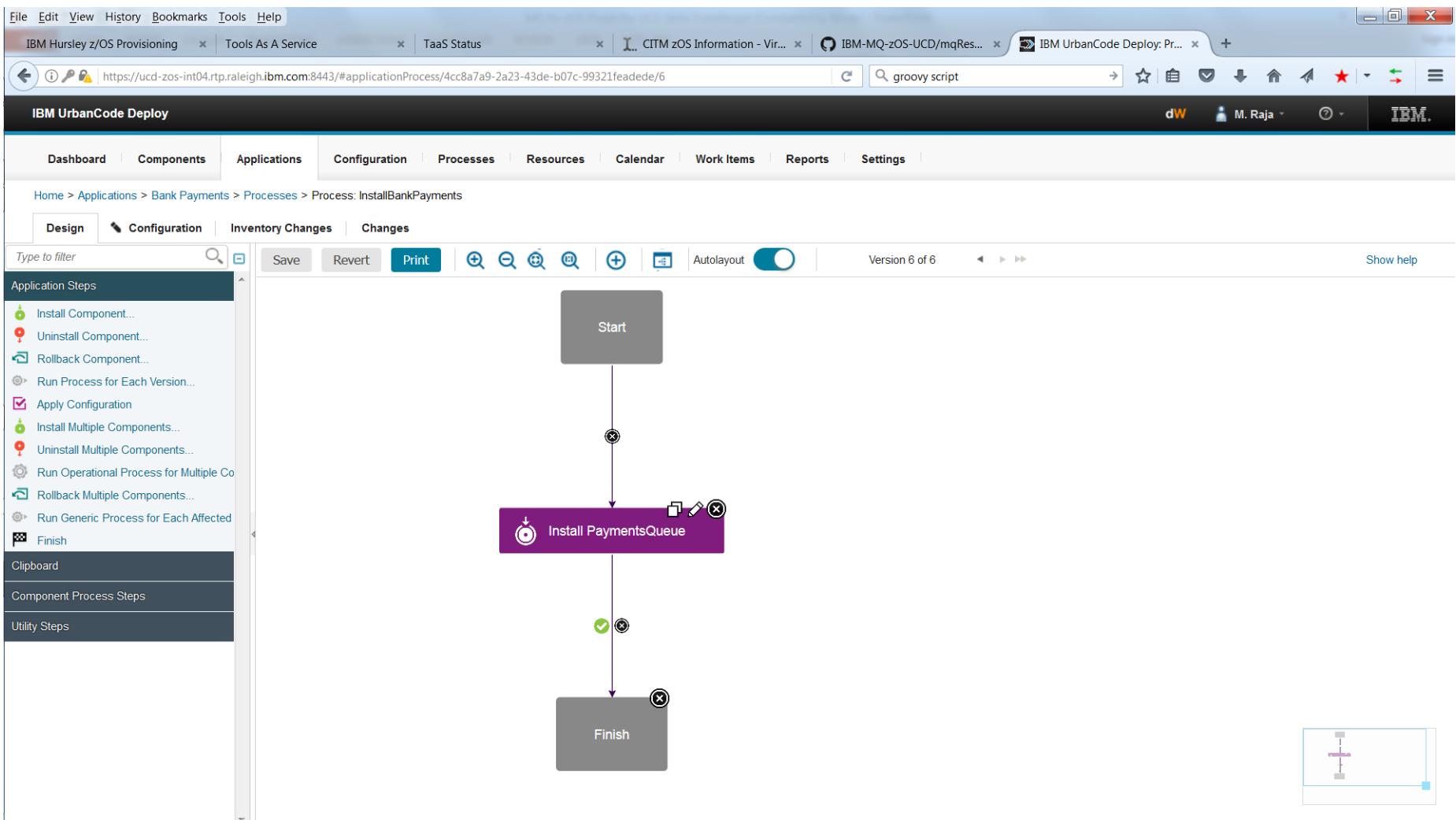
The screenshot shows a web browser window with multiple tabs open, including 'IBM Hursley z/OS Provisioning', 'Tools As A Service', 'TaaS Status', 'CITM zOS Information - Vir...', 'IBM-MQ-zOS-UCD/mqRes...', and 'IBM UrbanCode Deploy: B...'. The main content area is for the 'IBM UrbanCode Deploy' application, specifically the 'Bank Payments' application. The navigation bar includes 'Dashboard', 'Components', 'Applications', 'Configuration', 'Processes' (which is selected), 'Resources', 'Calendar', 'Work Items', 'Reports', and 'Settings'. Below this, a breadcrumb trail shows 'Home > Applications > Bank Payments'. The main view is titled 'Application: Bank Payments (show details)'. It features a 'Processes' tab selected, along with 'Environments', 'History', 'Configuration', 'Components', 'Blueprints', and 'Snapshots'. A prominent blue button labeled 'Create Process' is visible. A table lists three processes:

Process	Description	Actions
Clean all versions		Edit Copy Delete
InstallBankPayments	Install Bank Payments Component	Edit Copy Delete
InstallBankPaymentsZOSComponent		Edit Copy Delete

At the bottom left, it says '3 records - Refresh Print'. In the center, there is a page number '1 / 1'. On the right, there is a 'Rows' dropdown set to '10'.

Bank Payments Application – InstallBankPayments Process

Step installs component PaymentsQueue



Bank Payments Application – Install Process properties ..

Screenshot of the IBM UrbanCode Deploy interface showing the configuration of an application step.

The browser tabs at the top include: File, Edit, View, History, Bookmarks, Tools, Help; IBM Hursley z/OS Provisioning; Tools As A Service; TaaS Status; CITM zOS Information - Vir...; IBM-MQ-zOS-UCD/mqRes...; IBM UrbanCode Deploy: Pr...; groovy script.

The main navigation bar includes: Dashboard, Components, Applications, Configuration, Processes, Resources, Calendar, Work Items, Reports, Settings.

The breadcrumb path is: Home > Applications > Bank Payments > Processes > Process: InstallBankPayments.

The left sidebar lists Application Steps, Component Process Steps, and Utility Steps.

The central area shows the "Edit Properties" dialog for the "Install PaymentsQueue" step:

- Name: Install PaymentsQueue
- Component: PaymentsQueue
- Use Versions Without Status: Active
- Component Process: DefinePaymentsQueue
- Limit to Resource Tag: (dropdown menu)
- Maximum number of concurrent processes: -1
- Fail Fast:
- Precondition: 1

Buttons at the bottom of the dialog are OK and Cancel.

A large "Finish" button is visible at the bottom center of the page.

A small diagram in the bottom right corner shows a flowchart node with three outgoing arrows.

Bank Payments Application – InstallBankPayments process settings

The screenshot shows a web browser window with multiple tabs open, including 'IBM Hursley z/OS Provisioning', 'Tools As A Service', 'TaaS Status', 'IBM-MQ-zOS-UCD/mqRes...', 'IBM UrbanCode Deploy: Pr...', and 'groovy script'. The main content area is titled 'IBM UrbanCode Deploy' and shows the 'Processes' tab selected. The URL in the address bar is <https://ucd-zos-int04.rtp.raleigh.ibm.com:8443/#applicationProcess/4cc8a7a9-2a23-43de-b07c-99321feade/6/configuration>. The page displays the 'Basic Settings' for the 'InstallBankPayments' process. The 'Name' field is set to 'InstallBankPayments', 'Description' is 'Install Bank Payments Component', 'Inventory Management' is 'Automatic', 'Offline Agent Handling' is 'Check Before Execution', and 'Required Role' is empty. A note states: 'Version presets can be configured below. Whenever a value other than 'None' is given here, the user will not be able to provide a value at runtime, and any snapshot versions for that component will be ignored.' The 'Version for PaymentsQueue' dropdown is set to 'None'. At the bottom are 'Save' and 'Cancel' buttons.

Bank Payments Application – Components

The screenshot shows a web browser window with multiple tabs open, including 'IBM Hursley z/OS Provisioning', 'Tools As A Service', 'TaaS Status', 'CITM zOS Information - Vir...', 'IBM-MQ-zOS-UCD/mqRes...', 'IBM UrbanCode Deploy: B...', and 'groovy script'. The main content area is titled 'IBM UrbanCode Deploy' and shows the 'Components' tab selected under 'Applications'. The URL in the address bar is <https://ucd-zos-int04.rtp.raleigh.ibm.com:8443/#application/0a4d1f94-96f8-44d8-865e-686e455ddba0/components>. The page displays the 'Application: Bank Payments' with a '(show details)' link. Below this, there are tabs for 'Environments', 'History', 'Configuration' (which is selected), 'Blueprints', 'Snapshots', 'Processes', 'Calendar', and 'Changes'. A prominent 'Add Component' button is located at the top left of the component list. The component list itself has four categories: 'Failed Version Import' (0), 'Importing Version' (0), 'No Version' (0), and 'Successful' (2). The 'Successful' section contains two entries:

Name	Actions	Last Import	Last Version	Description
PaymentQueueZOS	Run Process Remove	Successful	v5	
PaymentsQueue	Run Process Remove	Successful	PaymentQ47	Payments Queue

At the bottom left, it says '2 records - Refresh Print'. In the bottom right corner, there are buttons for 'Rows' (set to 10) and a dropdown menu.

Bank Payments Application – PaymentsQueue properties

Screenshot of the IBM UrbanCode Deploy application interface showing the configuration for the PaymentsQueue component.

The browser tabs at the top include: File, Edit, View, History, Bookmarks, Tools, Help; IBM Hursley z/OS Provisioning; Tools As a Service; TaaS Status; CITM zOS Information - Vir...; IBM-MQ-zOS-UCD/mqRes...; IBM UrbanCode Deploy: P...; groovy script.

The main navigation bar includes: Dashboard, Components, Applications, Configuration, Processes, Resources, Calendar, Work Items, Reports, Settings.

The breadcrumb navigation shows: Home > Components > PaymentsQueue (show details).

The current view is under the Configuration tab, specifically the Basic Settings section.

Basic Settings

Component Properties

Environment Property Definitions

Resource Property Definitions

Version Property Definitions

Configuration File Templates

Version Import History

Basic Settings

Name *: PaymentsQueue

Description: Payments Queue

Teams: MQDemoTeam x +

Component Template: None

Component Type: Standard

Version Source Configuration

Source Configuration Type: RTC SCM

RTC Server URL *: https://ucd-zos-int04.rtp.raleigh.ibm.com:9443/ccm

RTC Username *: dave

RTC Password: *****

Stream *: MQ Dev Stream

Includes:
 **/*

Excludes:
 .

Bank Payments Application – PaymentsQueue properties

The screenshot shows the IBM UrbanCode Deploy application interface. The top navigation bar includes links for File, Edit, View, History, Bookmarks, Tools, and Help. Below the bar, several browser tabs are open, including "IBM Hursley z/OS Provisioning", "Tools As A Service", "TaaS Status", "CITM zOS Information - Vir...", "IBM-MQ-zOS-UCD/mqRes...", and "IBM UrbanCode Deploy: P...". The search bar contains the text "groovy script". The main header "IBM UrbanCode Deploy" is followed by a navigation menu with links for Dashboard, Components, Applications, Configuration, Processes, Resources, Calendar, Work Items, Reports, and Settings.

The current page path is Home > Components > PaymentsQueue. The title is "Component: PaymentsQueue (show details)". Below the title, there are tabs for Dashboard, Usage, Configuration (which is selected), Calendar, Versions, Processes, and Changes.

The left sidebar under "Basic Settings" has a tree view with "Component Properties" selected, and other options like Environment Property Definitions, Resource Property Definitions, Version Property Definitions, Configuration File Templates, and Version Import History.

The main content area is titled "Component Properties" and shows "Version 14 of 14". It features two buttons: "Add Property" and "Batch Edit". A table lists a single record:

Name	Value	Description	Actions
deploy.env.jobstatement	MAYURUTL JOB (ACCOUNT),'DEFAULT JOBCARD',CLASS=C, MSGCLASS=X,MSGLEVEL=(1,1),NOTIFY=&SYSUID	Job Card	Edit Delete

Below the table, it says "1 record - Refresh Print". To the right, there are navigation icons (back, forward, search) and a "Rows" dropdown set to 10.

Bank Payments Application – PaymentsQueue properties

The screenshot shows the IBM UrbanCode Deploy web interface. The top navigation bar includes links for File, Edit, View, History, Bookmarks, Tools, and Help. Below the bar, several tabs are open in the browser: IBM Hursley z/OS Provisioning, Tools As A Service, TaaS Status, CITM zOS Information - Vir..., IBM-MQ-zOS-UCD/mqRes..., and IBM UrbanCode Deploy P... . The main content area is titled "IBM UrbanCode Deploy" and shows the "Components" section. Under "Components", the "PaymentsQueue" component is selected. The "Configuration" tab is active, indicated by a blue border. The left sidebar contains links for Basic Settings, Component Properties, Environment Property Definitions (which is currently selected), Resource Property Definitions, Version Property Definitions, Configuration File Templates, and Version Import History. The main content area displays "Environment Property Definitions" with a sub-instruction: "Define properties here to be given values on each environment the component is used in." A large "Add Property" button is visible. Below this, a message says "Version 37 of 37" with navigation arrows. A table lists six environment properties:

Name	Label	Pattern	Required	Default Value	Description	Actions
jes.host	jes.host		false	WINMVS4C.hursley.ibm.com		Edit Delete
jes.user	jes.user		false	mayur		Edit Delete
jes.monitor.port	jes.monitor.port		false	16715		Edit Delete
jes.password	jes.password		false	****		Edit Delete
mqHighLevelQualifier	mqHighLevelQualifier		false	ANTZ.MQ.V000.CUR.OUT		Edit Delete
mqQueueManagerName	mqQueueManagerName		false	ZC03		Edit Delete

At the bottom of the table, it says "6 records - Refresh Print" and shows a page number "1 / 1". To the right, there is a "Rows" dropdown set to "10".

Bank Payments Application Payments Queue Component processes

The screenshot shows the IBM UrbanCode Deploy web interface. The browser tab bar includes 'File Edit View History Bookmarks Tools Help', 'IBM Hursley z/OS Provisioning', 'Tools As A Service', 'TaaS Status', 'CITM zOS Information - Vir...', 'IBM-MQ-zOS-UCD/mqRes...', and 'IBM UrbanCode Deploy: P...'. The search bar contains 'groovy script'. The main header has 'IBM UrbanCode Deploy' and the user 'M. Raja'. The navigation menu at the top includes 'Dashboard', 'Components', 'Applications', 'Configuration', 'Processes', 'Resources', 'Calendar', 'Work Items', 'Reports', and 'Settings'. Below this, a breadcrumb trail shows 'Home > Components > PaymentsQueue'. The main content area displays the 'Component: PaymentsQueue' page with a sub-header '(show details)'. A 'Create Process' button is visible. The 'Processes' tab is selected, showing a table with three rows:

Process	Description	Actions
DefinePaymentsQueue	Define Payments Queue to MQ	Edit Copy Delete
DefinePaymentsQueue (copy) (copy)	Define Payments Queue to MQ	Edit Copy Delete
Noop undeploy		Edit Copy Delete

At the bottom, there are links for '3 records - Refresh Print', a page navigation bar with '1 / 1', and a 'Rows' dropdown set to '10'.

Bank Payments Application - DefinePaymentsQueue process

Screenshot of the IBM UrbanCode Deploy interface showing the 'DefinePaymentsQueue' process flow.

The process flow is as follows:

```
graph TD; Start([Start]) --> Download[Download JSON representation of MQ resources (triplet of files)  
Download Artifacts (v. 28)]; Download --> Generate[Generate MQSC Commands  
Generate MQSC Commands (v. 1)  
PLUGIN DELETED]; Generate --> Submit[Submit MQ CSQUTIL Job  
Submit Job (v. 29)]; Submit --> Finish([Finish]);
```

Utility Steps (available in the sidebar):

- Finish
- Acquire Lock
- Release Lock
- Set Status
- Switch
- Join
- Note
- Manual Task
- Add Inventory Status
- Remove Inventory Status
- Run Component Process
- Run Generic Process

Clipboard (available in the sidebar):

- Application Server
- Cloud
- IBM UrbanCode Deploy
- Middleware
- Quality
- Repositories
- Resource Tools

Process Details (top right):

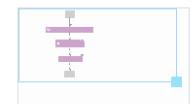
- Version 156 of 156
- Show help

Diagram Elements:

- Start:** Initial state node.
- Download:** Step to download MQ resources and artifacts.
- Generate:** Step to generate MQSC commands.
- Submit:** Step to submit the MQ CSQUTIL job.
- Finish:** Final state node.

Notes:

- 1) Download files from UCD repository
- 2) Generate MQSC commands
- 3) Submit CSQUTIL job to define MQ resources to target MQ for z/OS environment



Bank Payments Application - DefinePaymentsQueue process download step properties

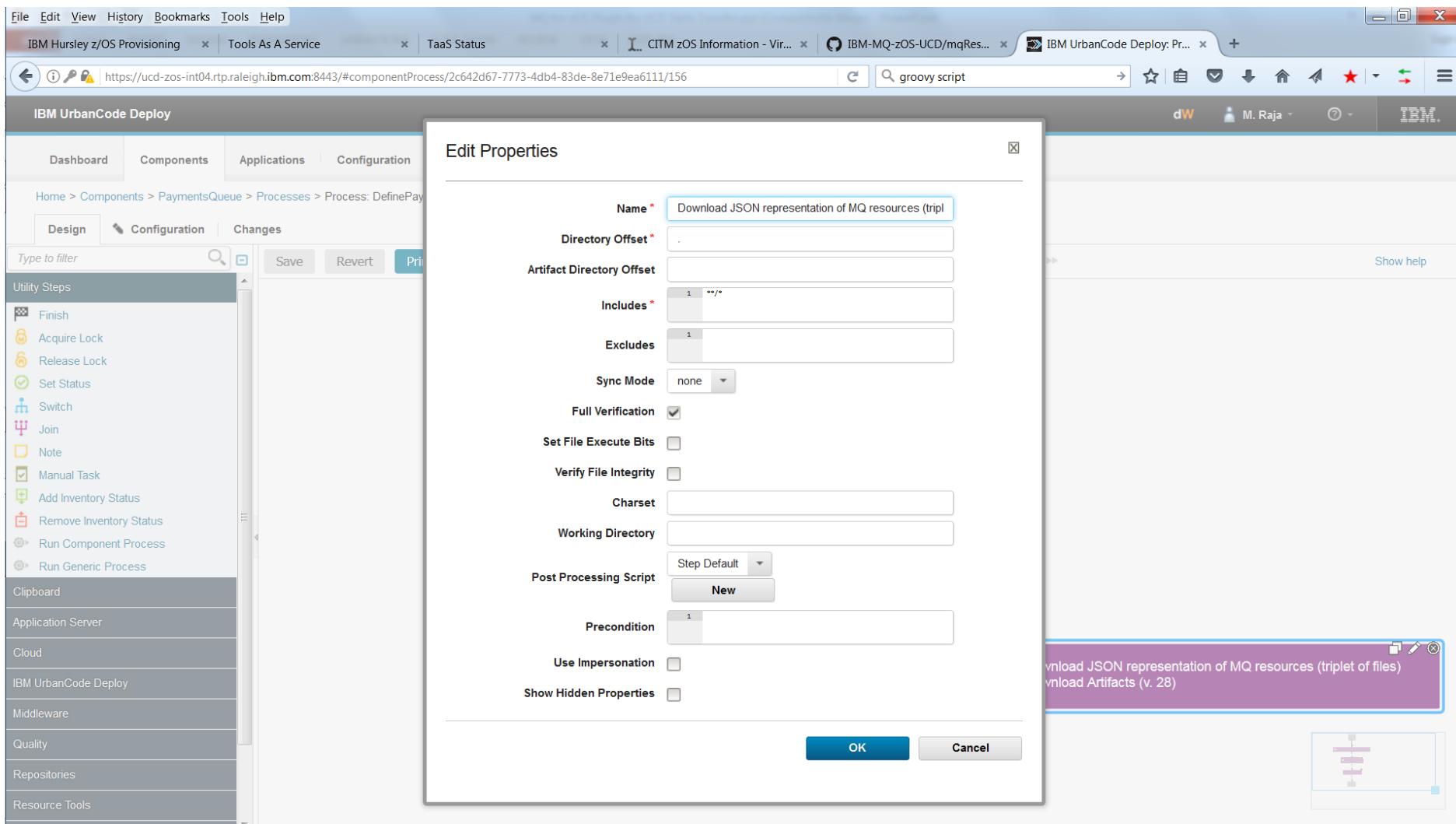
Screenshot of the IBM UrbanCode Deploy interface showing the configuration of a 'Download' step in a process named 'DefinePaymentsQueue'.

The 'Edit Properties' dialog is open, displaying the following configuration:

- Name:** Download JSON representation of MQ resources (triplet)
- Directory Offset:** .
- Artifact Directory Offset:** .
- Includes:** 1 **/*
- Excludes:** 1
- Sync Mode:** none
- Full Verification:** checked
- Set File Execute Bits:** unchecked
- Verify File Integrity:** unchecked
- Charset:** (empty field)
- Working Directory:** Step Default
- Post Processing Script:** New
- Precondition:** 1
- Use Impersonation:** unchecked
- Show Hidden Properties:** unchecked

A tooltip at the bottom right of the dialog provides a summary of the step's function:

Download JSON representation of MQ resources (triplet of files)
Download Artifacts (v. 28)



Bank Payments Application - DefinePaymentsQueue process generate MQSC step properties

Screenshot of the IBM UrbanCode Deploy interface showing the configuration of a process step.

The browser tabs include: File, Edit, View, History, Bookmarks, Tools, Help; IBM Hursley z/OS Provisioning; Tools As A Service; TaaS Status; CITM zOS Information - Vir...; IBM-MQ-zOS-UCD/mqRes...; IBM UrbanCode Deploy: Pr...; groovy script.

The main navigation bar includes: Dashboard, Components, Applications, Configuration, Resources, Calendar, Work Items, Reports, Settings.

The current path is: Home > Components > PaymentsQueue > Processes > Process: DefinePaymentsQueue.

The left sidebar lists Utility Steps: Finish, Acquire Lock, Release Lock, Set Status, Switch, Join, Note, Manual Task, Add Inventory Status, Remove Inventory Status, Run Component Process, Run Generic Process.

The Design tab is selected, showing Configuration and Changes.

The configuration dialog is titled "Edit Properties".

Properties defined:

- Name: Generate MQSC Commands
- File Sub Folders:
 - 1. ZKEBankApplication-hQ/mqdef/PQ
 - 2. ZKEBankApplication-hQ/mqdef/no
- Base File Name Filter: *.mqdef_base
- Overrides File Type: mqdef_overrides
- Properties File Type: mqdef_properties
- Working Directory: (empty)
- Post Processing Script: Step Default (New button)
- Precondition: 1
- Use Impersonation: (unchecked)
- Show Hidden Properties: (checked)
- Target Environment Name: \${p:environment.name}
- User Data: \${p:component.name} \${p:version.name}
- Enable/Disable Trace: (checked)

Buttons at the bottom: OK, Cancel, Finish.

A message box in the bottom right corner says: "Generate MQSC Commands" and "Generate MQSC Commands (v. 1) PLUGIN DELETED".

Bank Payments Application - DefinePaymentsQueue process submit CSQUTIL job step properties

The screenshot shows the IBM UrbanCode Deploy interface. The left sidebar lists components like Applications, Configuration, Design, Configuration, and Changes. Under Utility Steps, several options are listed: Finish, Acquire Lock, Release Lock, Set Status, Switch, Join, Note, Manual Task, Add Inventory Status, Remove Inventory Status, Run Component Process, and Run Generic Process. The Applications section is currently selected.

The main area displays a process named "DefinePaymentsQueue". A modal dialog titled "Edit Properties" is open, showing configuration for a job step. The "Name" field is set to "Submit MQ CSQUTIL Job". The "JCL Dataset" and "JCL File" fields are empty. The "JCL" code block contains the following JCL:

```
1 //IOSCDIS EXEC PGM=CSQUTIL,PARM='${mqQueueManagerName}',CO
2 //SYSPRINT DD SYSOUT=
3 //STEPLIB DD DSE ${mqHighLevelQualifier}.SCSQNAME,DISP=SH
4 //           DD DSE ${mqHighLevelQualifier}.SCSQUTH,DISP=SH
5 //SYSIN DD *
6   COMMAND DOMAIN(CDS)
7 /*

1 //WAVRUTL JOB (ACCOUNT), 'DEFAULT JOBCARD',CLASS=C,
2 //           MSGCLASS=X,MSGLEVEL=(1,1),NOTIFY=&SYSPID
```

The "Default Job Statement" and "Replace Tokens" sections are empty. The "Replace Token sets for Each Job" section also has one entry. The "Wait For Job" and "Stop On Fail" checkboxes are checked. The "Timeout" field is set to 60. The "Show Output" field is set to ALL, and the "Max Lines" field is set to 1000. The "Max Return Code" field is set to 4. The "Working Directory" and "Post Processing Script" fields are empty. A "Step Default" dropdown is set to "New".

A purple callout box in the bottom right corner indicates a successful submission: "Submit MQ CSQUTIL Job Submit Job (v. 29)".

Bank Payments Application

PaymentsQueue, DefinePaymentsQueue ..

The screenshot shows a web browser window with multiple tabs open, including 'IBM Hurstley z/OS Provisioning', 'Tools As A Service', 'TaaS Status', 'CITM zOS Information - Vir...', 'IBM-MQ-zOS-UCD/mqRes...', and 'IBM UrbanCode Deploy: Pr...'. The main content area is titled 'IBM UrbanCode Deploy' and shows the 'Components' section. Under 'Components', 'PaymentsQueue' is selected, and under 'Processes', 'Process: DefinePaymentsQueue' is being configured. The 'Configuration' tab is active. The 'Basic Settings' panel on the left lists 'Component Process Properties'. The 'Basic Settings' form on the right contains the following fields:

Name *	DefinePaymentsQueue
Description	Define Payments Queue to MQ
Process Type *	Deployment
Inventory Status *	Active
Default Working Directory *	<code>\$(p:resource/work.dir)/\${p:component.name}</code>
Required Role	(dropdown menu)

At the bottom of the form are 'Save' and 'Cancel' buttons.

Bank Payments Application

Running the PaymentsQueue process

The screenshot shows the IBM UrbanCode Deploy web interface. The top navigation bar includes links for File, Edit, View, History, Bookmarks, Tools, and Help. Below the bar, several tabs are open in the browser: IBM Hursley z/OS Provisioning, Tools As A Service, TaaS Status, CITM zOS Information - Vir..., IBM-MQ-zOS-UCD/mqRes..., IBM UrbanCode Deploy: Pr..., and Service Details - IBM Blue... . The search bar contains the text "groovy script". The main content area is titled "IBM UrbanCode Deploy" and shows the "Processes" tab selected. The breadcrumb navigation indicates the current location: Home > Components > PaymentsQueue > Process Request on PaymentsQueue. The main title is "Deployment of Component: PaymentsQueue" with a "(show details)" link. Below this, there are two tabs: "Log" (selected) and "Properties". The "Execution" section displays a table of steps:

Step	Progress	Start Time	Duration	Status
1. Download JSON representation of MQ resources (triplet of files)		10:32:45 AM	0:00:04	Running
2. Generate MQSC Commands				Not Started
3. Submit MQ CSQUTIL Job				Not Started
Total Execution	0 / 3	10:32:45 AM	0:00:04	Running

Buttons for "Pause", "Cancel", and "Download All Logs" are located above the table. On the right side of the table, there are "Expand All" and "Collapse All" buttons.

Bank Payments Application

Running the PaymentsQueue process ..

The screenshot shows the IBM UrbanCode Deploy web interface. The top navigation bar includes links for File, Edit, View, History, Bookmarks, Tools, and Help. Below the bar, a search bar contains the text "groovy script". The main content area displays the "IBM UrbanCode Deploy" interface with tabs for Dashboard, Components, Applications, Configuration, Processes, Resources, Calendar, Work Items, Reports, and Settings. The current view is under the Components tab, specifically for the "PaymentsQueue" component. A breadcrumb trail indicates the path: Home > Components > PaymentsQueue > Process Request on PaymentsQueue. The title "Deployment of Component: PaymentsQueue" is followed by a "(show details)" link. Below this, there are two tabs: "Log" (selected) and "Properties". The "Execution" section shows a timeline from 10:32:45 AM to 10:33:08 AM. The timeline is divided into five segments: Start (10:32:45 AM), Progress (3 / 3), Status (Success), Duration (0:00:22), and End (10:33:08 AM). A gear icon is positioned next to the start time. A checkbox labeled "Dock timeline at top" is checked. Below the timeline, there is a "Download All Logs" button and links for "Expand All" and "Collapse All". A detailed table below lists the steps of the process:

Step	Progress	Start Time	Duration	Status
1. Download JSON representation of MQ resources (triplet of files)	::	10:32:45 AM	0:00:09	Success
2. Generate MQSC Commands	::	10:32:54 AM	0:00:07	Success
3. Submit MQ CSQUTIL Job	::	10:33:02 AM	0:00:05	Success
Total Execution	3 / 3	10:32:45 AM	0:00:22	Success

Bank Payments Application

Example output log

The screenshot shows the IBM UrbanCode Deploy interface with the "Output Log" tab selected. The log window displays the command-line output for a component process request. The log content is as follows:

```
Working Directory /MV4C/udeployagentr/opt/ibm-ucd/agent/var/work//PaymentsQueue
=====
1 plugin: MQ for zOS, id: com.ibm.ucd.plugin.mqzos, version: 1
2 plugin command: '/MV4C/udeployagentr/opt/ibm-ucd/agent/opt/groovy-1.8.8/bin/groovy' '-cp' '/MV4C/udeployagentr/opt/ibm-ucd/agent/var/plugins/com.ibm.ucd.plugin.mqzos_1_32f3abc43ae5d7572d03c831439f986dde6e23639fe9e0a33eeaa0bea778cf3d8
3 working directory: /MV4C/udeployagentr/opt/ibm-ucd/agent/var/work/PaymentsQueue
4 properties:
5   PLUGIN_INPUT_PROPS=/MV4C/udeployagentr/opt/ibm-ucd/agent/var/temp/logs1645061358840673887/input.props
6   PLUGIN_OUTPUT_PROPS=/MV4C/udeployagentr/opt/ibm-ucd/agent/var/temp/logs1645061358840673887/output.props
7   baseFileNameFilter=.*.mqdef_base
8   fileSubFolders=JKEBankApplication-MQ/mqdef/PQ
9   JKEBankApplication-MQ/mqdef/MQ
10  overridesFileType=mqdef_overrides
11  propertiesFileType=mqdef_properties
12  targetEnvironmentName=Test
13  traceEnabled=true
14  userData=PaymentsQueue PaymentQ47
15 environment:
16   AGENT_HOME=/MV4C/udeployagentr/opt/ibm-ucd/agent
17   AH_AUTH_TOKEN=****
18   AH_WEB_URL=https://ucd-zos-int04.rtp.raleigh.ibm.com:8443
19   AUTH_TOKEN=****
20   DS_AUTH_TOKEN=****
21   DS_SYSTEM_ENCODING=IBM-1047
22   JAVA_OPTS=-Dfile.encoding=IBM-1047 -Dconsole.encoding=IBM-1047
23   PLUGIN_HOME=/MV4C/udeployagentr/opt/ibm-ucd/agent/var/plugins/com.ibm.ucd.plugin.mqzos_1_32f3abc43ae5d7572d03c831439f986dde6e23639fe9e0a33eeaa0bea778cf3d8
24   PLUGINDIALOGUE_ID=96c90252-7805-4dd6-a0ec-9808eb5bd870
25   WE_ACTIVITY_ID=b4cbfd41-55ed-4440-aeb6-774d5b13a62b
26 =====
27 Entry: generate_commands.groovy
28 Data: fileSubFolders: JKEBankApplication-MQ/mqdef/PQ
29 JKEBankApplication-MQ/mqdef/MQ
30 Data: baseFileNameFilter: .*_.mqdef_base
31 Data: overridesFileType: mqdef_overrides
32 Data: propertiesFileType: mqdef_properties
33 Data: targetEnvironmentName: Test
34 Data: userData: PaymentsQueue PaymentQ47
35 =====
```

At the bottom of the log window, there is a "Download Log" button and a page navigation bar showing page 1 of 6. The browser address bar shows the URL: https://ucd-zos-int04.rtp.raleigh.ibm.com:8443/#componentProcessRequest/61228498-0c6d-49be-bc60-67eda8108bc5/log.

Bank Payments Application

Sample Trace and INFORMATION messages

The screenshot shows the IBM UrbanCode Deploy interface with the following details:

- Header:** File, Edit, View, History, Bookmarks, Tools, Help.
- Tabs:** IBM Hursley z/OS Provisioning, Tools As A Service, TaaS Status, CITM zOS Information - Vir..., IBM-MQ-zOS-UCD/mqRes..., IBM UrbanCode Deploy: Pr..., Service Details - IBM Blue... (active tab).
- Address Bar:** https://ucd-zos-int04.rtp.raleigh.ibm.com:8443/#componentProcessRequest/61228498-0c6d-49be-bc60-67eda8108bc5/log
- Left Sidebar:** IBM UrbanCode Deploy, Home > Components Deployment, Log (selected), Properties, Execution (Start: 10:32:45 AM), Download All Log, Step (1. Download, 2. Generate, 3. Submit M), Total Execution.
- Central Content:** Output Log window titled "Output Log".
 - Working Directory:** /MV4C/udeployagentr/opt/ibm-ucd/agent/var/work//PaymentsQueue
 - Log Content:** A large block of log entries showing the execution of a Groovy script to generate MQ definitions. The log includes entries for generating commands, setting base file paths, and defining MQ properties like mqdef_base, mqdef_overrides, and mqdef_properties. It also shows the creation of paymentQ.queue and paymentQ.tx queue definitions.
 - Buttons:** Download Log, a page number indicator (1 / 6), and navigation arrows.
- Right Sidebar:** Dock timeline at top (checkbox checked), End: 10:33:08 AM, Timeline bar showing progress, Expand All, Collapse All, and a list of access steps.
- Bottom Navigation:** time, Highlight All, Match Case, Whole Words, 2 of 2 matches.

Sample Triplet Files

payQ - Base File

```
1  {
2      "resource": {
3          "queue": [
4              {
5                  "command": "DEFINE",
6                  "name": "PAYMENT.QUEUE",
7                  "type": "local",
8                  "general": {
9                      "description": "Queue for bank payments"
10                 },
11                 "storage": {
12                     "maximumDepth": 1000,
13                     "maximumMessageLength": 32000
14                 },
15                 "extended": {
16                     "allowSharedInput": true
17                 },
18                 "trigger": {
19                     "enabled": false,
20                     "type": "none"
21                 }
22             ],
23             "channel": [
24                 {
25                     "command": "DEFINE",
26                     "name": "@RQMGR@.TO.@LQMGR@",
27                     "type": "receiver"
28                 }
29             }
30         }
31     }
```

Note: I found Notepad++ to be a good editor for JSON as it matches brackets and also highlights the matching pair if you place the cursor next to a bracket. But, I am sure there are better editors available.

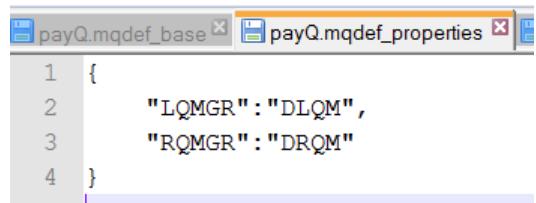


payQ - Overrides File

```
payQ.mqdef_base payQ.mqdef_properties payQ.mqdef_overrides
1  {
2      "resource": {
3          "queue": [
4              {
5                  "command": "DEFINE",
6                  "name": "PAYMENT.QUEUE",
7                  "type": "local",
8                  "deploymentTargets": [
9                      {
10                         "targetEnvName": "Dev",
11                         "deploy": true,
12                         "storage": {
13                             "maximumDepth": 10000,
14                             "maximumMessageLength": 16384
15                         }
16                     },
17                     {
18                         "targetEnvName": "Test",
19                         "deploy": true,
20                         "storage": {
21                             "maximumDepth": 20000,
22                             "maximumMessageLength": 32768
23                         }
24                     },
25                     {
26                         "targetEnvName": "QA",
27                         "deploy": false,
28                         "storage": {
29                             "maximumDepth": 30000,
30                             "maximumMessageLength": 65536
31                         }
32                     },
33                     {
34                         "targetEnvName": "PROD",
35                         "deploy": false
36                     }
37                 ],
38             }
39         ]
40     }
41     ]
42   }
43   ]
44   ]
45   ]
46   ]
47   ]
48   ]
49   ]
50   ]
51   ]
52   ]
53   ]
54   ]
55   ]
56   ]
57   ]
58 }
```

```
    ...
    "channel": [
        {
            "command": "DEFINE",
            "name": "@RQMGR@.TO.@LQMGR@",
            "type": "receiver",
            "deploymentTargets": [
                {
                    "targetEnvName": "Dev",
                    "deploy": true,
                    "extended": {
                        "useDeadLetterQueue": true
                    }
                },
                {
                    "targetEnvName": "Test",
                    "deploy": true,
                    "general": {
                        "descr": "Receiver Channel for Test Environment",
                        "maximumMessageLength": "32768"
                    }
                }
            ]
        }
    ]
}
```

payQ - Properties File



```
1 {  
2     "LQMGR": "DLQM",  
3     "RQMGR": "DRQM"  
4 }
```

Note: Sample triplet files can be found at:

https://github.com/IBM-UrbanCode/IBM-MQ-zOS-UCD/tree/master/doc/sample_triplet_files

Other samples are documented in the whitepaper which can be found at:

https://github.com/IBM-UrbanCode/IBM-MQ-zOS-UCD/tree/master/doc/white_paper

Source Code

Source Code

IBM MQ z/OS plugin for IBM UrbanCode Deploy used to Generate MSQC Commands. <https://developer.ibm.com/urbancode/p...>

urbancode urbancode-deploy ibm-mq zos

24 commits

1 branch

2 releases

3 contributors

Branch: master ▾

New pull request

Find file

Clone or download ▾

 gurugg Update examples

Latest commit ac05fee on 13 Apr

 doc	Update examples	6 months ago
 license	Standarized to other OSS Plugins	7 months ago
 plugin	Spelling	7 months ago
 src	Rename mqscutil.groovy to MQSCUtil.groovy	7 months ago
 .gitignore	Updated Travis Support	7 months ago
 .travis.yml	Updated Travis Support	7 months ago
 README.md	Updated zip location	7 months ago
 build.xml	Explode groovy-plugin-utils into the classes folder	7 months ago
 ivy.xml	Explode groovy-plugin-utils into the classes folder	7 months ago

Code – plugin.xml file – defines the plugin properties

Branch: master ▾

IBM-MQ-zOS-UCD / plugin / plugin.xml

Find file Copy path

 nhmathis Bumped version

697d08c on 15 Mar

1 contributor

70 lines (69 sloc) | 4.17 KB

Raw Blame History   

```
1  <?xml version="1.0" encoding="UTF-8"?>
2  <!--
3      © Copyright IBM Corporation 2016, 2017.
4      This is licensed under the following license.
5      The Eclipse Public 1.0 License (http://www.eclipse.org/legal/epl-v10.html)
6      U.S. Government Users Restricted Rights: Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
7  -->
8  <plugin xmlns="http://www.urbancode.com/PluginXMLSchema_v1" xmlns:server="http://www.urbancode.com/PluginServerXMLSchema_v1" xmlns:
9      <header>
10         <identifier id="com.ibm.ucd.plugin.mqzos" name="MQ for zOS" version="2"/>
11         <description> The MQ for zOS plug-in includes deployment activities for MQ for zOS
12         </description>
13         <tag>Middleware/Messaging/MQ for zOS</tag>
14     </header>
15     <step-type name="Generate MQSC Commands">
16         <description>Generate MQSC commands from the base file, overrides file and property file.</description>
17         <properties>
18             <property name="fileSubFolders" required="true">
19                 <property-ui description="Specify a list of sub folders that contain the base, overrides and properties files. Specify each
20                               default-value="" />
21             </property>
22             <property name="baseFileNameFilter" required="true">
23                 <property-ui description="Specify a base file name filter using a regular expression. For example: *.mqdef_base for base f
24                               default-value=".*.mqdef_base" />
25             </property>
```

Note: The exact syntax of this XML file is defined by UCD.



Code – generate_commands.groovy

IBM-UrbanCode / IBM-MQ-zOS-UCD

Watch 8 Star 1 Fork 1

Code Issues 0 Pull requests 0 Projects 0 Insights

Branch: master IBM-MQ-zOS-UCD / src / main / scripts / generate_commands.groovy Find file Copy path

nhmathis Rename MQSCUtil c07ef1b on 3 Mar

1 contributor

281 lines (244 sloc) | 12.5 KB Raw Blame History

```
1 /**
2  * Copyright IBM Corporation 2016, 2017.
3  * This is licensed under the following license.
4  * The Eclipse Public 1.0 License (http://www.eclipse.org/legal/epl-v10.html)
5  * U.S. Government Users Restricted Rights: Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
6 */
7
8 import com.urbancode.air.AirPluginTool
9 import com.urbancode.air.plugin.mqzos.MQSCUtil
10
11 // generate_commands.groovy is the first method (groovy script) that is invoked when
12 // the Generate MQSC Commands plugin step is run as part of an IBM UCD process. It:
13 //
14 // - Reads and validates the input variables,
15 // - Validates that the required files exist,
16 // - Creates the mqsc resource definitions property variable that contains the MQSC commands,
17 // - Writes the user data (set to Component and Version names by default) to the mqsc resource definitions property variable.
18 // - For all triplet of files in the work directory that meet the base file name filter criteria,
19 //   It verifies that the files exist, creates an instance of class MQSCUtil and invokes method
20 //   GenerateMQSCFormDefintions to generate the MQSC form of MQ commands for the REST form of MQ
21 //   Commands represented in the triplet of files, and to write them to the mqsc resource definitions property variable.
22 // - Displays details of the number of base definition files (and hence triplets of files) processed.
23 // - If no files are processed, a suitable exception is thrown.
24 // - If any exceptions are thrown, terminate with an return code of 1 (FAIL).
```

A good summary of what this method does.



Code – generate_commands.groovy – new mqscutil

```
214 // Write the names of the triplet of files about to be read as comments to the MQSC resource
215 // If no commands are generated from the files, it is not a problem as only comments will be
216 // Users may find it useful to know that files were read but no commands were generated for
217 // resource definition error, or that residual files have not been cleaned up.
218 mqscResourceDefinitions += '\n*\n* ' + baseFileNameAndType + ', ' + overridesFileNameAndType
219
220
221 // Call MQSCUtil to generate MQSC commands based on the 3 input files
222 mqscResourceDefinitions = new MQSCUtil().generateMQSCFormDefinitions(
223     baseFile,
224     new File(fullWorkDirPath, overridesFileNameAndType),
225     new File(fullWorkDirPath, propertiesFileNameAndType),
226     mqscResourceDefinitions,
227     targetEnvironmentName,
228     traceEnabled)
229
230 // Increment count of base files in sub directory processed.
231 countOfBaseFilesInSubDirectoryProcessed ++
232 // Increment count of base files processed.
233 countOfTotalBaseFilesProcessed ++
234 }
235
236 println "*** INFORMATION: ' + countOfBaseFilesInSubDirectory + ' base definition files found in sub "
237 println "*** INFORMATION: ' + countOfBaseFilesInSubDirectoryProcessed + ' base definition files proc
238
239 if (countOfBaseFilesInSubDirectoryProcessed == 0) {
240     println 'WARNING: Folder ' + subDirectory + ' does not contain any base files so no files w
241 }
```



Code – mqscutil.groovy

Branch: master ▾

[Find file](#) [Copy path](#)

[IBM-MQ-zOS-UCD](#) / [src](#) / [main](#) / [groovy](#) / [com](#) / [urbancode](#) / [air](#) / [plugin](#) / [mqzos](#) / [MQSCUtil.groovy](#)



nhmathis Rename mqscutil.groovy to MQSCUtil.groovy

7f3b4c3 on 15 Mar

1 contributor

1018 lines (856 sloc) | 42.2 KB

Raw Blame History

```
1  /**
2   * Copyright IBM Corporation 2016, 2017.
3   * This is licensed under the following license.
4   * The Eclipse Public 1.0 License (http://www.eclipse.org/legal/epl-v10.html)
5   * U.S. Government Users Restricted Rights: Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
6  */
7
8 package com.urbancode.air.plugin.mqzos
9
10 import groovy.json.JsonSlurper
11
12 /**
13  * MQSC util class to generate MQSC form JSON form resource definitions specified in a base,
14  * overrides and properties file. An MQ resource attributes mappings file is used to facilitate
15  * the conversion from the JSON file to the MQSC representation.
16 */
17 class MQSCUtil {
18
19     // Trace flag.
20     def private trace
21
22     // Character set encoding for the platform where the UCD agent is running.
23     def private charSetName
24
25     /**
26      * Generates MQSC form resource definitions from JSON form resource definitions specified in a base,
27      * overrides and properties file.
28      *
29      * @param baseFile - file containing the base MQ resource definitions (in JSON form).
30      * @param overridesFile - file containing the override values for MQ resource definitions (in JSON form).
31      * @param propertiesFile - file containing property values (in JSON form) that are to be used to replace tokens in the MQ
```

Summary of what this class does.

Methods are reasonably modular and indicate what they do

Code – mqResourceAttributes.mapping

Branch: master ▾

IBM-MQ-zOS-UCD / src / main / scripts / resourceMappings / mqResourceAttributes.mappings

Find file Copy path



nhmathis Standardized to other OSS Plugins

37b9070 on 2 Mar

1 contributor

809 lines (809 sloc) | 23.3 KB

Raw Blame History

```
1  {
2      "resource": {
3          "queue": {
4              "optionalParms": {
5                  "commandScope": {
6                      "attrDataType": "string",
7                      "attrMapping": "CMDSCOPE"
8                  },
9                  "like": {
10                     "attrDataType": "string",
11                     "attrMapping": "LIKE"
12                     },
13                     "noReplace": {
14                         "attrDataType": "flag",
15                         "attrMapping": "NOREPLACE"
16                         },
17                         "force": {
18                             "attrDataType": "flag",
19                             "attrMapping": "FORCE"
20                             },
21                             "keepAuthorityRecords": {
22                                 "attrDataType": "keyword",
23                                 "attrMapping": "AUTHREC",
24                                 "attrValueMappings": {
```

Mapping from JSON format
to MQSC format



Tables in whitepaper also show JSON to MQSC mappings

The following table lists the REST attribute groups, the REST attributes and the MQSC attributes they map to, the REST attribute values and the corresponding MQSC attribute values, for queues. This information is required when defining queue resource representations in the base and overrides files (see examples below):

REST Attribute Group Name	REST Attribute Name	MQSC Attribute Name	REST Attribute Value	MQSC Attribute Value
<u>optionalParms</u>	<u>commandScope</u>	CMDSCOPE	<i>String</i>	<i>string</i>
	<u>like</u>	LIKE	<i>String</i>	<i>string</i>
	<u>noReplace</u>	NOREPLACE	<u>noReplace</u>	NOREPLACE
	<u>force</u>	FORCE	Force	FORCE
	<u>keepAuthorityRecords</u>	AUTHOREC	<u>keepAuthorityRecords</u>	NO
	<u>purge</u>	PURGE	Purge	PURGE
	<u>queueSharingGroupDisposition</u>	QSGDISP	Copy	COPY
			Group	GROUP
			Private	PRIVATE
			<u>Qmgr</u>	QMGR
			shared	SHARED
<u>alias</u>	<u>targetName</u>	TARGET	<i>string</i>	<i>string</i>
	<u>targetType</u>	TARGTYPE	queue	QUEUE
			topic	TOPIC
<u>applicationDefaults</u>	<u>clusterBind</u>	DEFBIND	<u>notFixed</u>	NOTFIXED
			<u>onOpen</u>	OPEN
			<u>onGroup</u>	GROUP
	<u>messagePropertyControl</u>	PROPCtl	all	ALL
			compatible	COMPAT
			force	FORCE
			none	NONE
			version6Compatible	V6COMPAT
	<u>messagePersistence</u>	DEFPSIST	<u>nonPersistent</u>	NO
			persistent	YES
	<u>messagePriority</u>	DEFPRTY	<i>int</i>	<i>int</i>

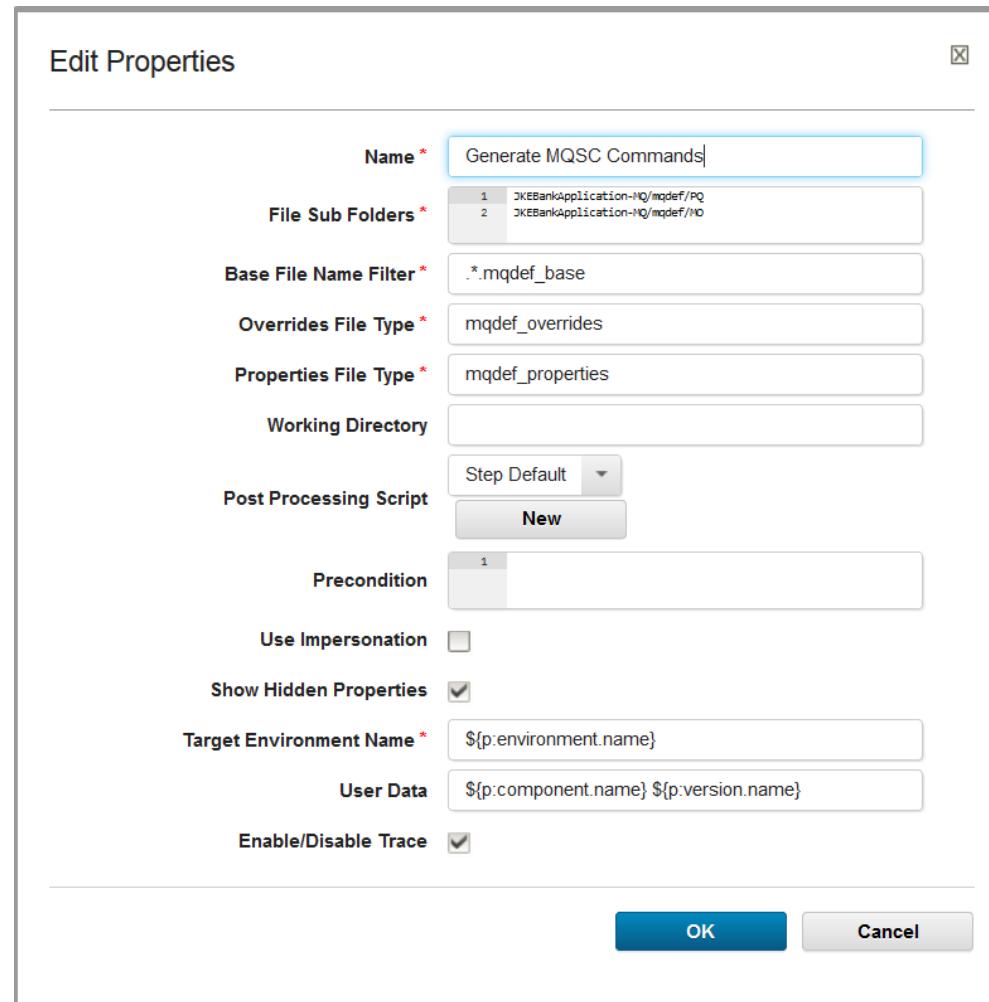
Note:

- 1) This is a subset of the queue attribute mappings.
- 2) There is also a table for the channel attribute mappings.

Trace

Trace – Enable/Disable

- Trace is enabled from the plugin step properties in the UCD process
- Entry, Data and Exit trace entries are produced (see live demo)
- println's are also issued to the log output file for the plugin step (see live demo)



Enable/Disable trace

Bank Payments Application

Sample Trace and INFORMATION messages

The screenshot shows the IBM UrbanCode Deploy interface with the following details:

- Header:** File, Edit, View, History, Bookmarks, Tools, Help.
- Tabs:** IBM Hursley z/OS Provisioning, Tools As A Service, TaaS Status, CITM zOS Information - Vir..., IBM-MQ-zOS-UCD/mqRes..., IBM UrbanCode Deploy: Pr..., Service Details - IBM Blue... (active tab).
- Address Bar:** https://ucd-zos-int04.raleigh.ibm.com:8443/#componentProcessRequest/61228498-0c6d-49be-bc60-67eda8108bc5/log
- Search Bar:** one wheel bike in china
- Left Sidebar:** IBM UrbanCode Deploy, Home > Components Deployment, Log (selected), Properties, Execution (Start 10:32:45 AM), Step (1. Download, 2. Generate, 3. Submit M), Total Execution.
- Main Content:** Output Log titled "Working Directory /MV4C/udeployagentr/opt/ibm-ucd/agent/var/work/PaymentsQueue". The log output is as follows:

```
28 Entry: generate_commands.groovy
29 Data: fileSubFolders: JKEBankApplication-MQ/mqdef/PQ
30 JKEBankApplication-MQ/mqdef/MQ
31 Data: baseFileNameFilter: *.mqdef_base
32 Data: overridesFileType: mqdef_overrides
33 Data: propertiesFileType: mqdef_properties
34 Data: targetEnvironmentName: Test
35 Data: userData: PaymentsQueue PaymentQ47
36 Data: fullWorkDirPath: /MV4C/udeployagentr/opt/ibm-ucd/agent/var/work/PaymentsQueue/JKEBankApplication-MQ/mqdef/PQ
INFORMATION: MQ definition base file: paymentQ.mqdef_base
INFORMATION: MQ definition overrides file: paymentQ.mqdef_overrides
INFORMATION: MQ definition properties file: paymentQ.mqdef_properties
40 Entry: generateMQSCFormDefinitions
41 Data: baseFile:
/MV4C/udeployagentr/opt/ibm-ucd/agent/var/work/PaymentsQueue/JKEBankApplication-MQ/mqdef/PQ/paymentQ.mqdef_base
Data: overridesFile:
/MV4C/udeployagentr/opt/ibm-ucd/agent/var/work/PaymentsQueue/JKEBankApplication-MQ/mqdef/PQ/paymentQ.mqdef_overrides
Data: propertiesFile:
/MV4C/udeployagentr/opt/ibm-ucd/agent/var/work/PaymentsQueue/JKEBankApplication-MQ/mqdef/PQ/paymentQ.mqdef_properties
Data: mqscResourceDefinitions:
*
* User Data: PaymentsQueue PaymentQ47
*
*
* paymentQ.mqdef_base, paymentQ.mqdef_overrides, paymentQ.mqdef_properties
*
54 Data: targetEnvironmentName: Test
55 Entry: parseFile
56 Data: fileName: /MV4C/udeployagentr/opt/ibm-ucd/agent/var/work/PaymentsQueue/JKEBankApplication-MQ/mqdef/PQ/paymentQ.mqdef_base
Data: parsedData: [resource:[queue:[optionalParms:[noReplace:noReplace, commandScope:], alias:[targetName:PL00.@ENVMQ0.FT5_IDL_TX.RQ, targetType:queue], r
59 Exit : parseFile
60 Entry: buildOverridesMap
61 Entry: parseFile
62 Data: fileName: /MV4C/udeployagentr/opt/ibm-ucd/agent/var/work/PaymentsQueue/JKEBankApplication-MQ/mqdef/PQ/paymentQ.mqdef_overrides
```
- Bottom Navigation:** Download Log, Page Number 1 / 6.
- Bottom Filter:** time, Highlight All, Match Case, Whole Words (2 of 2 matches).

Bank Payments Application

Sample Trace and INFORMATION messages

```
INFORMATION: MQ definition base file: paymentQ.mqdef_base
INFORMATION: MQ definition overrides file: paymentQ.mqdef_overrides
INFORMATION: MQ definition properties file: paymentQ.mqdef_properties
Entry: generateMQSCFormDefinitions
  Data: baseFile:
/MV4C/udeployagentr/opt/ibm-ucd/agent/var/work/PaymentsQueue/JKEBankApplication-MQ/mqdef/PQ/paymentQ.mqdef_base
  Data: overridesFile:
/MV4C/udeployagentr/opt/ibm-ucd/agent/var/work/PaymentsQueue/JKEBankApplication-MQ/mqdef/PQ/paymentQ.mqdef_overrides
  Data: propertiesFile:
/MV4C/udeployagentr/opt/ibm-ucd/agent/var/work/PaymentsQueue/JKEBankApplication-MQ/mqdef/PQ/paymentQ.mqdef_properties
  Data: mqscResourceDefinitions:
*
* User Data: PaymentsQueue PaymentQ47
*
*
* paymentQ.mqdef_base, paymentQ.mqdef_overrides, paymentQ.mqdef_properties
*

  Data: targetEnvironmentName: Test
Entry: parseFile
  Data: fileName: /MV4C/udeployagentr/opt/ibm-ucd/agent/var/work/PaymentsQueue/JKEBankApplication-MQ/mqdef/PQ/paymentQ.mqdef_base
  Data: parsedData: [resource:[queue:[[[optionalParms:[noReplace:noReplace, commandScope:]], alias:[targetName:PL00.@ENVMQ@.FT5_IDL_TX.RQ, targetType:queue], Exit : parseFile
Entry: buildOverridesMap
Entry: parseFile
  Data: fileName: /MV4C/udeployagentr/opt/ibm-ucd/agent/var/work/PaymentsQueue/JKEBankApplication-MQ/mqdef/PQ/paymentQ.mqdef_overrides
```

Further Information

Further Information on MQ plugins for UCD

- For details about the MQ for Distributed UCD plugin, see:

<https://developer.ibm.com/urbancode/plugin/websphere-mq/>

- For details about the MQ for z/OS UCD plugin and white paper, see:

<https://developer.ibm.com/urbancode/plugin/ibm-mq-for-zos/>

- For a Blogs about the plugin, see:

https://www.ibm.com/developerworks/community/blogs/messaging/entry/IBM_MQ_for_z_OS_Generate_MQSC_Commands?lang=en

<https://developer.ibm.com/urbancode/2017/04/17/ibm-mq-zos-generate-mqsc-commands-plugin-ibm-ucd/>

Further information on z/OSMF Workflows for middleware

- <http://www.redbooks.ibm.com/redpapers/pdfs/redp5416.pdf>
- https://www.ibm.com/support/knowledgecenter/en/SSFKSJ_9.0.0/com.ibm.mq.doc/q019925.htm
- https://www.ibm.com/developerworks/community/blogs/messaging/entry/MQzOS_online-click-provisioning?lang=en
- https://www.ibm.com/developerworks/community/blogs/messaging/entry/Using_zOSMF_V2_2_Cloud_Provisioning_to_stand_up_MQ_resources?lang=en
- For a full list of the workflows and services provided for MQ, CICS, DB2, IMS, WAS (classic and Liberty Profile Server), Click on the Cloud tab at:
<http://www-03.ibm.com/systems/z/os/zos/features/zosmf/>

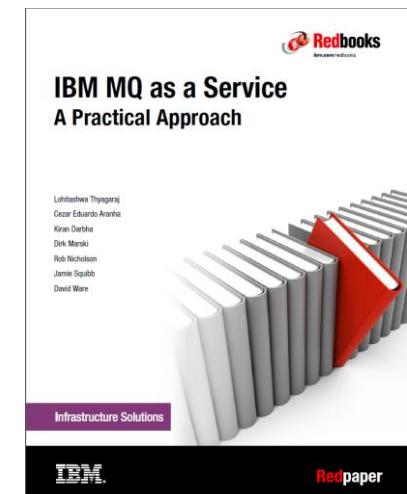
Further information on z/OS Provisioning Toolkit

- See a step by step guide at:

<https://developer.ibm.com/mainframe/2017/01/10/provisioning-a-mq-development-environment-with-the-ibm-zos-provisioning-toolkit/>

Redbooks - IBM MQ as a Service

- Why MQ as a Service
- How to discover user requirements for an automated system
- Technologies used to build self-service environments
- Includes examples of:
 - UCD and IBM-MQ plugin



Summary

- What is IBM Urban Code Deploy ?
- MQ Provisioning Technology
- MQ Plugins for UCD
- What do you need to run the plugin ?
 - UCD Server, UCD Agent, UCD Artifacts
- Supported Software Levels
- Does the MQ for z/OS plugin run on distributed platforms ?
- MQ Resource Deployments
- Example Bank Payments Application
- Sample Triplet Files
- Source Code
- Trace
- Further Information
- Q&A

Copyright and Trademarks

© IBM Corporation 2017. All Rights Reserved.

IBM, the IBM logo, and ibm.com 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 IBM trademarks is available on the Web at “Copyright and trademark information” at www.ibm.com/legal/copytrade.shtml.



Notice and disclaimers

- Copyright © 2017 by International Business Machines Corporation (IBM). No part of this document may be reproduced or transmitted in any form without written permission from IBM.
- **U.S. Government Users Restricted Rights — use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM.**
- Information in these presentations (including information relating to products that have not yet been announced by IBM) has been reviewed for accuracy as of the date of initial publication and could include unintentional technical or typographical errors. IBM shall have no responsibility to update this information. **This document is distributed “as is” without any warranty, either express or implied. In no event shall IBM be liable for any damage arising from the use of this information, including but not limited to, loss of data, business interruption, loss of profit or loss of opportunity.** IBM products and services are warranted according to the terms and conditions of the agreements under which they are provided.
- IBM products are manufactured from new parts or new and used parts. In some cases, a product may not be new and may have been previously installed. Regardless, our warranty terms apply.”
- **Any statements regarding IBM's future direction, intent or product plans are subject to change or withdrawal without notice.**
- Performance data contained herein was generally obtained in a controlled, isolated environments. Customer examples are presented as illustrations of how those customers have used IBM products and the results they may have achieved. Actual performance, cost, savings or other results in other operating environments may vary.
- References in this document to IBM products, programs, or services does not imply that IBM intends to make such products, programs or services available in all countries in which IBM operates or does business.
- Workshops, sessions and associated materials may have been prepared by independent session speakers, and do not necessarily reflect the views of IBM. All materials and discussions are provided for informational purposes only, and are neither intended to, nor shall constitute legal or other guidance or advice to any individual participant or their specific situation.
- It is the customer's responsibility to insure its own compliance with legal requirements and to obtain advice of competent legal counsel as to the identification and interpretation of any relevant laws and regulatory requirements that may affect the customer's business and any actions the customer may need to take to comply with such laws. IBM does not provide legal advice or represent or warrant that its services or products will ensure that the customer is in compliance with any law.

Notice and disclaimers continued

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products in connection with this publication and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products. IBM does not warrant the quality of any third-party products, or the ability of any such third-party products to interoperate with IBM's products. **IBM expressly disclaims all warranties, expressed or implied, including but not limited to, the implied warranties of merchantability and fitness for a particular purpose.**

The provision of the information contained herein is not intended to, and does not, grant any right or license under any IBM patents, copyrights, trademarks or other intellectual property right.

IBM, the IBM logo, ibm.com, AIX, BigInsights, Bluemix, CICS, Easy Tier, FlashCopy, FlashSystem, GDPS, GPFS, Guardium, HyperSwap, IBM Cloud Managed Services, IBM Elastic Storage, IBM FlashCore, IBM FlashSystem, IBM MobileFirst, IBM Power Systems, IBM PureSystems, IBM Spectrum, IBM Spectrum Accelerate, IBM Spectrum Archive, IBM Spectrum Control, IBM Spectrum Protect, IBM Spectrum Scale, IBM Spectrum Storage, IBM Spectrum Virtualize, IBM Watson, IBM z Systems, IBM z13, IMS, InfoSphere, Linear Tape File System, OMEGAMON, OpenPower, Parallel Sysplex, Power, POWER, POWER4, POWER7, POWER8, Power Series, Power Systems, Power Systems Software, PowerHA, PowerLinux, PowerVM, PureApplication, RACF, Real-time Compression, Redbooks, RMF, SPSS, Storwize, Symphony, SystemMirror, System Storage, Tivoli, WebSphere, XIV, z Systems, z/OS, z/VM, z/VSE, zEnterprise and zSecure are trademarks of International Business Machines Corporation, registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at: www.ibm.com/legal/copytrade.shtml.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both. Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.