Managing Service Bus Namespaces & Artifacts

Alan Smith

ACTIVE SOLUTION

@alansmith   www.cloudcasts.net

Overview

Namespaces, Queues, Topics & Subscriptions

Managing Artifacts with the Azure Portal

Demo: Managing Service Bus Artifacts with the Azure Portal
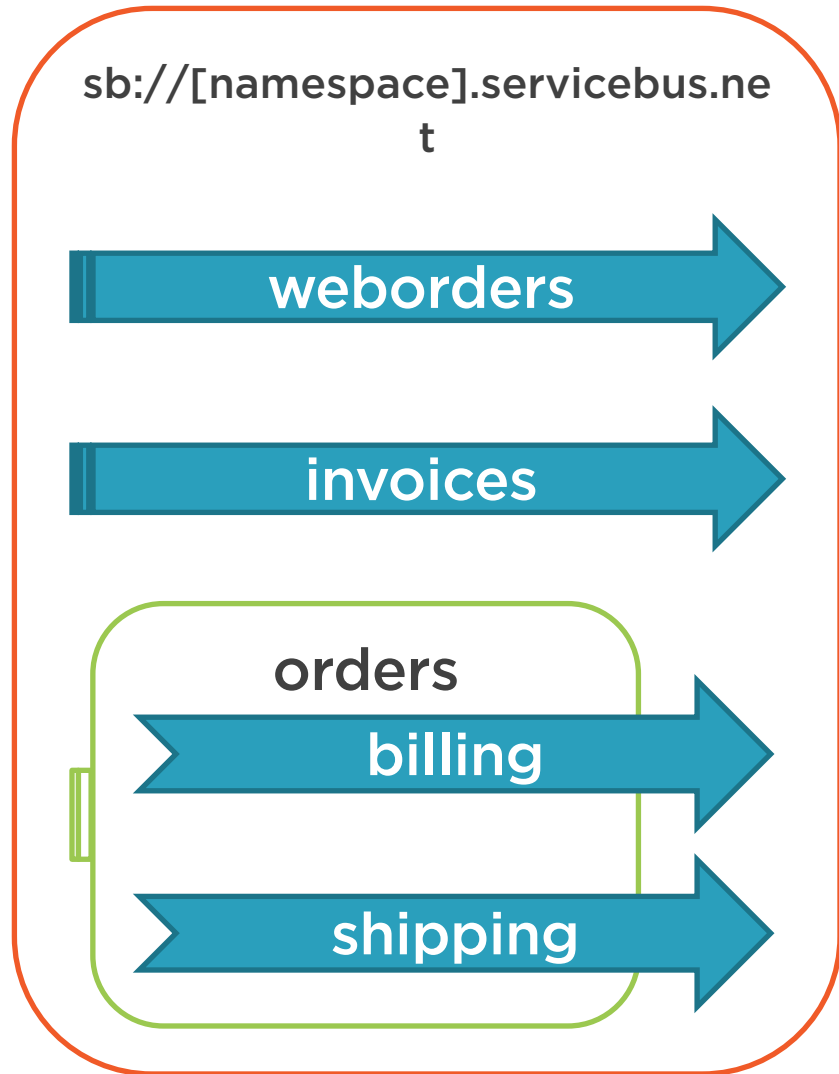
Managing Artifacts with Code

Demo: Managing Service Bus Artifacts with Code

Managing Artifacts with Command-line Tools

Demo: Managing Artifacts with Azure CLI

Namespaces, Queues, Topics & Subscriptions

Service Bus Namespace

sb://[namespace].servicebus.net

weborders

invoices

orders

billing

shipping

Created in a specified data center

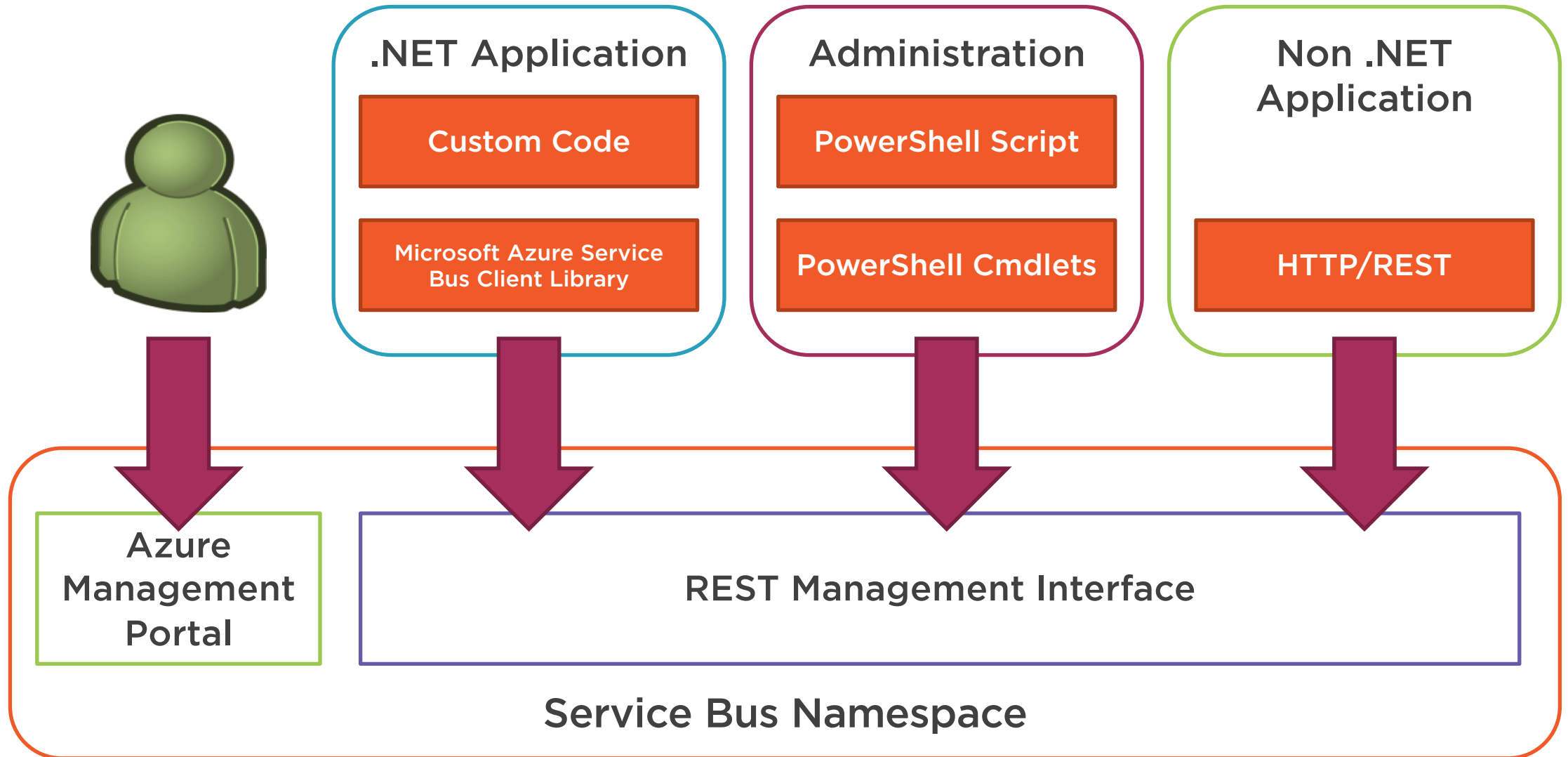## Must be unique across all data centers

Container for messaging artifacts

- Queues
- Topics
  - Act as a container for subscriptions

Forms root namespace for artifacts

- sb:[namespace]/servicebus.net/weborders
- sb:[namespace]/servicebus.net/invoices
- sb:[namespace]/servicebus.net/orders

Management Interfaces

.NET Application
- Custom Code
- Microsoft Azure Service Bus Client Library

Administration
- PowerShell Script
- PowerShell Cmdlets

Non .NET Application
- HTTP/REST

Azure Management Portal

REST Management Interface

Service Bus Namespace

Messaging Entity Properties

Property	Queue	Topic	Subscription
Path	Yes	Yes	
Name			Yes
DefaultMessageTimeToLive	Yes	Yes	Yes
DuplicateDetectionHistoryTimeWindow	Yes	Yes	
MaxSizeInMegabytes	Yes	Yes	
RequiresDuplicateDetection	Yes	Yes	
EnableDeadLetteringOnMessageExpiration	Yes		Yes
LockDuration	Yes		Yes
MaxDeliveryCount	Yes		Yes
RequiresSession	Yes		Yes
EnableDeadLetteringOnFilterEvaluationExceptions			Yes

Sending Properties

Property	Description
DefaultMessageTimeToLive	Specifies the time that enqueue messages will be valid for before expiring if not explicitly specified in the message properties.
RequiresDuplicateDetection	Specifies if messages sent with the same message ID will be ignored.
DuplicateDetectionHistoryTimeWindow	Duration to store message ID values for duplicate detection.
MaxSizeInMegabytes	Specifies the storage capacity of the entity.

Receiving Properties

Property	Description
EnableDeadLetteringOnMessageExpiration	Specifies whether messages will be moved to the dead-letter queue when their time to live expires.
LockDuration	Specifies the time interval a message will remain in the locked state after it has been revived using the peek-lock mode.
MaxDeliveryCount	The number of times a message can be received before being dead-lettered.
RequiresSession	Specifies whether message sessions will be used on the messaging entity.

Managing Artifacts with the Azure Portal

Managing Artifacts with the Azure Portal

Managing Artifacts with the Azure Portal

Demo

Managing Artifacts with the Azure Portal
- – Creating a Service Bus namespace
- – Creating a queue
- – Creating topics and subscriptions

Managing Artifacts with Code

Useful Classes

Class	Description
ManagementClient	Provides methods for creating, listing and deleting queues, topics and subscriptions.
QueueDescription TopicDescription SubscriptionDescription	Used to specify the properties of queues, topics and subscriptions.

NamespaceManager Methods

Method	Description
CreateQueueAsync, CreateTopicAsync, CreateSubscriptionAsync	Creates a new queue/topic/subscription in the service namespace with the given path.
DeleteQueueAsync, DeleteTopicAsync, DeleteSubscriptionAsync	Deletes the queue/topic/subscription described by the path relative to the service namespace base address.
GetQueueAsync, GetTopicAsync, GetSubscriptionAsync	Retrieves a queue/topic/subscription from the service namespace.
GetQueuesAsync, GetTopicsAsync, GetSubscriptionsAsync	Retrieves an enumerable collection of all queues/topics/subscriptions in the service namespace.
QueueExistsAsync, TopicExistsAsync, SubscriptionExistsAsync	Determines whether a queue/topic/subscription exists in the service namespace.

Creating Queues

```csharp
// Create a queue with default properties
public async Task<QueueDescription> CreateQueueAsync(string path)

// Create a queue with custom properties
public async Task<QueueDescription> CreateQueueAsync(QueueDescription description)
```

Created within service bus namespace

Path must be unique within namespace

Property defaults overridden using description class

Creating Queues Example

```csharp
// Create a custom queue description.
QueueDescription customDescription = new
    QueueDescription("customproperties")
{
    DefaultMessageTimeToLive = TimeSpan.FromHours(1),
    EnableDeadLetteringOnMessageExpiration = true,
    RequiresDuplicateDetection = true,
    DuplicateDetectionHistoryTimeWindow = TimeSpan.FromMinutes(5),
    LockDuration = TimeSpan.FromMinutes(2),
    RequiresSession = true
};

// Create a queue using the queue description.
QueueDescription customQueueDescription =
    await managementClient.CreateQueueAsync(customDescription);
```

Creating Topics

```csharp
// Create a queue with default properties
public async Task<TopicDescription> CreateTopicAsync(string path)


// Create a queue with custom properties
public async Task<TopicDescription> CreateTopicAsync(TopicDescription description)
```

- **Created within service bus namespace**

- **Path must be unique within namespace**

- **Property defaults overridden using description class**

Adding Subscriptions to Topics

```csharp
// Create a topic with default properties
public async Task<SubscriptionDescription> CreateSubscriptionAsync
        (string path, string name)


// Create a topic with custom properties
public async Task<SubscriptionDescription> CreateSubscriptionAsync
        (SubscriptionDescription description)
```

- **Added to a topic**

- **Name must be unique within topic**

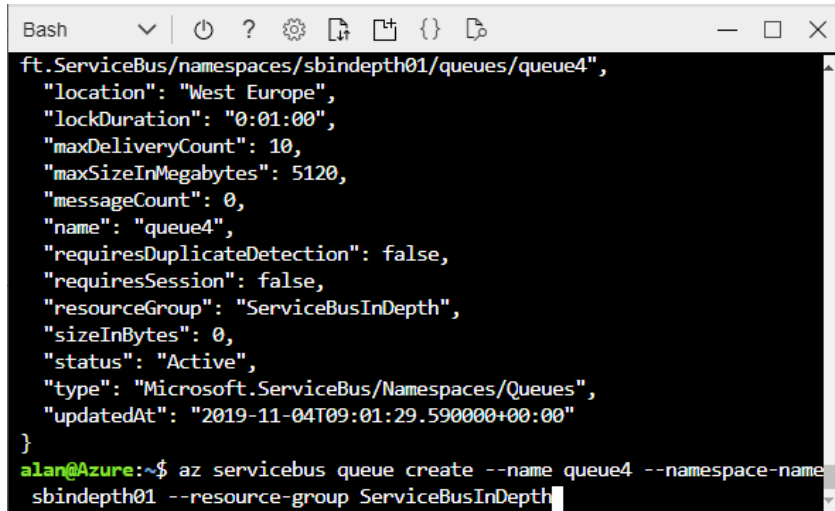- **Property defaults overridden using description class**

Demo

Managing Artifacts with Code

- Authenticating with ManagementClient

- Creating Queues, Topics and Subscriptions

- Managing Queues, Topics and Subscriptions

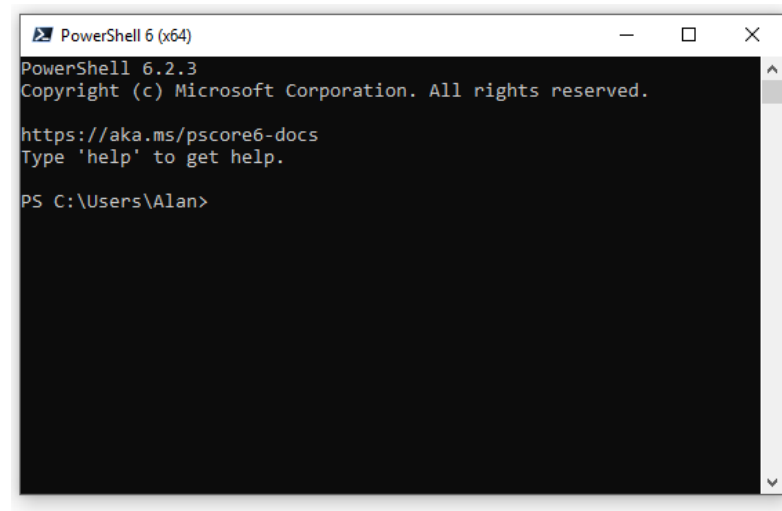- Setting properties of Queues, Topics and Subscriptions

Managing Artifacts with Command-line Tools

Azure Command-line Management

Azure Command-line Interface (CLI)

Azure PowerShell

```
az servicebus namespace create
    --resource-group ServiceBusInDepth
    --name sbindepth01
    --location westeurope
```

Azure CLI – Create Service Bus namespace

```
az servicebus queue create
    --name queue1
    --namespace-name sbindepth01
    --resource-group ServiceBusInDepth
```

Azure CLI – Create queue

```
az servicebus topic create
    --name topic1
    --namespace-name sbindepth01
    --resource-group ServiceBusInDepth
```

Azure CLI – Create topic

```
az servicebus topic subscription create
    --name subscription1
    --topic-name topic1
    --namespace-name sbindepth01
    --resource-group ServiceBusInDepth
```

Azure CLI – Create subscription

Demo

Managing Artifacts with Azure CLI

- Creating a Service Bus Namespace
- Creating Queues
- Creating Topics and Subscriptions
- Leverage CLI help and examples

Summary

Messaging entities are contained in a Service Bus namespace

- Queues
- Topics & Subscriptions

Azure Portal provides easy but basic management

Microsoft Azure Service Bus SDK provides classes to manage messaging entities

Command-line tools can be used for automated provisioning

- Azure CLI
- PowerShell