

# Exchange 2007 Resource Manager

---

## Table of Contents

Credits	1
Overview	1
Exchange Resource/Consumer model	2
Basic Resources	2
Business Organization Resource/Consumer model	3
Consumer Resource/Consumer model	5
Allocation and Resource Sharing Algorithms	6
Typical Scenario: Life cycle of a Mailbox in RM	8
Note:	9
AddExchangeResource	9
AllocateOrganization	10
AllocateMailbox	13
DeallocateMailbox	15
DeallocateOrganization	15
RemoveExchangeResources	16
Glossaries	17

## Credits

<b>Author:</b>	<b>Arthur Fang(arthfa), Shanming Xuan(shaxuan)</b>
<b>Technical Reviewer:</b>	<b>Linden Goffar (lindeng), Julie Xu(juliexu), Steve Li(shali)</b>
<b>UA Reviewer:</b>	<b>John Dong(jodong)</b>

## Overview

Hosted Exchange Services is now facing the challenge of managing larger and more complex Exchange resources (mail store, public store, mailbox, public folder, and front-end server) while tracking the kind of information available in Hosted Exchange Services. Managing Exchange resources means not only managing the data size and availability but also the enforcing the hosting

policies with a very good understanding of how existing storage is used. This allows for sound strategic planning and prompt response to organization changes.

Exchange 2007 Resource Manager is an integrated tool. This builds a Resource/ Consumer model with a set of algorithms upon MPF namespace to manipulate [CoreRMO](#) and [BlockRMO](#) providers (Figure 1). Exchange 2007 Resource Manager lets Hosted Exchange Services to understand, control, and manage the quantity and type of data that is stored on Exchange servers. By using Exchange 2007 Resource Manager, administrators can put quotas on mailboxes and public folders, screen mail and public stores, and generate comprehensive storage reports. This advanced instrument not only helps Hosted Exchange Services to efficiently monitor the existing storage resources, but also helps planning and implementation of future organization changes.



Figure 1 Logical Architecture

## Exchange Resource/Consumer model

There are three business models in HMC: Business organization, IIW and Consumer model. Business organization model and IIW model are similar in functionality. Therefore, we only discuss Business organization model and consumer model in this document. According to HMC business model, Exchange 2007 Resource Manager builds two types of Resource/Consumer models that are based on underlying providers CoreRMO and BlockRMO.

### Basic Resources

There are 3 major types of resources in Exchange 2007 Resource Manager:

#### MailStore

MailStore is a database to hold Exchange mailboxes. Exchange 2007 Resource Manager treats MailStore as a basic management unit in Hosted E-mail Service.

The only character of a MailStore is its capacity. MailStore capacity is determined by total capacity and free capacity. When administrator creates a MailStore, Exchange 2007 Resource

Manager creates a MailStore instance in CoreRMO and set its capacity in BlockRMO. When allocation occurs, Exchange 2007 Resource Manager re-calculates its free capacity.

### PublicStore

PublicStore is a database to hold Public Folders and their contents. PublicStore is also a basic management unit in Hosted E-Mail Service.

The only character of a PublicStore is its capacity. PublicStore capacity is determined by total capacity and free capacity. When administrator adds a new PublicStore into hosted environment, Exchange 2007 Resource Manager creates a PublicStore instance in CoreRMO and set its capacity in BlockRMO. When allocation occurs, Exchange 2007 Resource Manager re-calculates its free capacity.

### Front-end server

This is a legacy resource type.

- **ExchFrontEndServer07**  
Front-end server is not a typical resource type. It cannot be consumed by any consumers. Exchange 2007 Resource Manager only records front-end servers in CoreRMO.

## Business Organization Resource/Consumer model

### Resources

- **ExchMailStore07**  
ExchMailStore07 is a type of MailStore consumed by Business organization. The basic unit of ExchMailStore07 is Mega Bytes.
- **ExchPublicStore07**  
ExchPublicStore07 is a type of PublicStore consumed by Business organization. The basic unit of ExchPublicStore07 is Mega Bytes.

### Derived Resources

There are also two types of Derived Resources for Business Organization:

- **ExchOwningOrgMailResource07**  
Mail storage owned by an organization. Figure 2 illustrates the procedure of creating an ExchOwningOrgMailResource07 resource. The basic unit of ExchOwningOrgMailResource07 is Kilo Bytes.

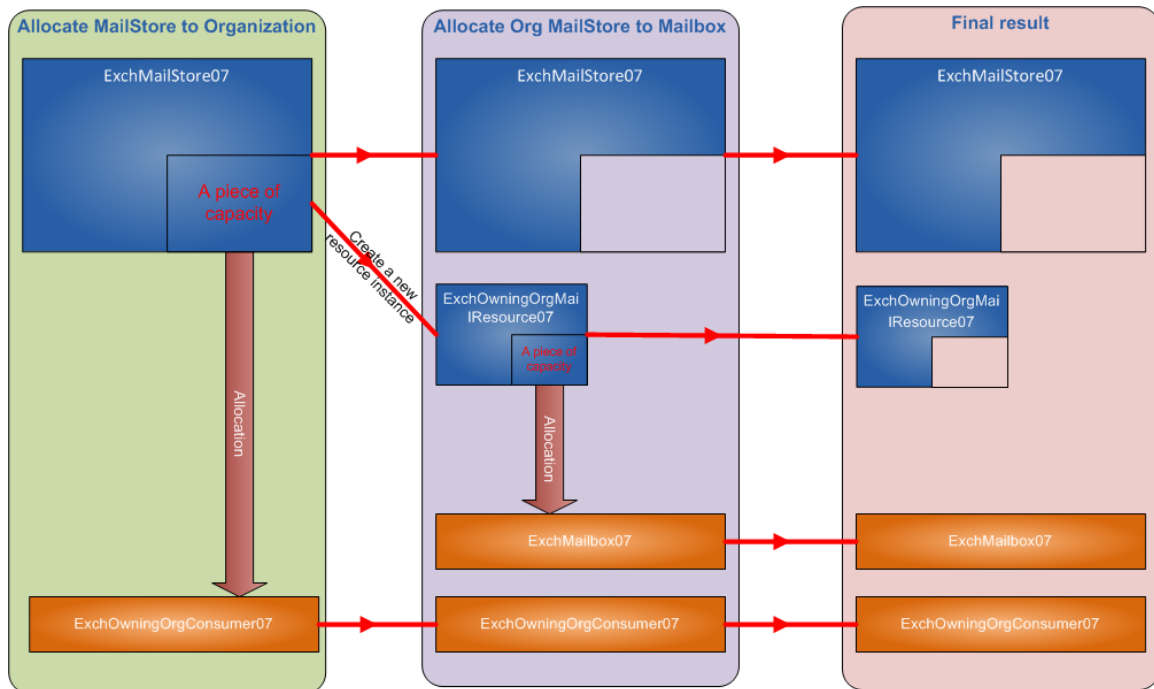


Figure 2 Business Organization Resource/Consumer Model

1. A piece of capacity of resource ExchMailStore07 instance is allocated to an [ExchOwningOrgConsumer07](#) instance.
  2. An ExchOwningOrgMailResource07 instance is created in CoreRMO.
  3. Set the capacity of ExchOwningOrgMailResource07 instance according to the allocated capacity from ExchMailStore07 instance.
- **ExchOwningOrgPublicResource07**  
Public folder storage owned by an organization. ExchOwningOrgPublicResource07 has the same logic as ExchOwningOrgMailResource07. The basic unit of ExchOwningOrgPublicResource07 is Mega Bytes.

## Consumers

- **ExchOwningOrgConsumer07**  
An ExchOwningOrgConsumer07 instance represents an organization. ExchOwningOrgConsumer07 instances can consume two types of resources: ExchMailStore07 and ExchPublicStore07. One ExchOwningOrgConsumer07 instance can consume multiple ExchMailStore07 instances and ExchPublicStore07 instances. After an ExchOwningOrgConsumer07 instance consumes an ExchMailStore07 instance, an ExchOwningOrgMailResource07 instance is created. The ExchOwningOrgConsumer07 instance and ExchOwningOrgMailResource07 instance have the same instance name. This indicates their relationship. The same logic happens when an ExchOwningOrgConsumer07 instance consumes ExchPublicStore07 instance.
- **ExchMailbox07**

An ExchMailbox07 instance represents a mailbox used by a user in the organization. After an ExchOwningOrgConsumer07 instance consumes ExchMailStore07 instances, an ExchMailbox07 instance (the mailbox of a user in the organization) can consume generated ExchOwningOrgMailResource07 instance.

An ExchMailbox07 instance can only consume one ExchOwningOrgMailResource07 instance.

- **ExchPublicFolder07**

An ExchPublicFolder07 instance represents a public folder used by an organization. After an ExchOwningOrgConsumer07 instance consumes ExchPublicStore07 instances, an ExchPublicFolder07 instance (a public folder used by the organization) can consume generated ExchOwningOrgPublicResource07 instances.

An ExchPublicFolder07 instance can only consume one ExchOwningOrgPublicResource07 instance.

- **ExchShareHogType07**

This is an assistant consumer type. Only one ExchShareHogType07 instance exists in Resource Manager and it is created in the initialization of Exchange 2007 Resource Manager. For more information, please read [Resource Sharing](#).

## Consumer Resource/Consumer model

### Resource

- **ExchConsumerMailStore07**

The customers of ExchConsumerMailStore07 are consumers. The basic unit of ExchConsumerMailStore07 is Kilo Bytes.

### Consumer

- **ExchConsumerMailbox07**

An ExchConsumerMailbox07 instance represents a consumer.

An ExchOwningOrgConsumer07 instance can only consume one ExchConsumerMailStore07 instance.

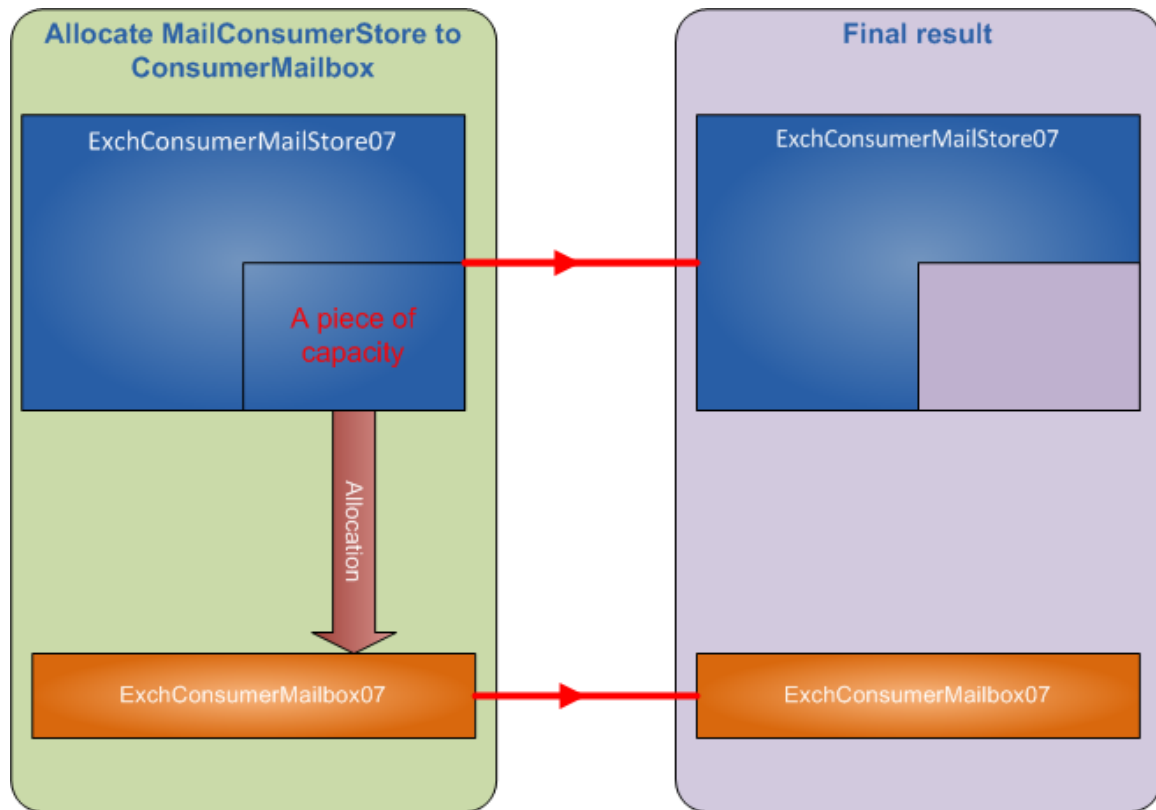


Figure 3 Consumer Resource/Consumer model

## Allocation and Resource Sharing Algorithms

### Allocation Algorithms

There are two allocation algorithms:

- **Standard Allocation**

The standard allocation can be described as fixed quota allocation.

When an organization subscribes an Exchange Service Plan as standard allocation (dynamicQuota = false), the capacity of resources the organization can consume is determined. Exchange 2007 Resource manager creates an ExchOwningOrgConsumer07 instance and allocates ExchMailStore07 instances to it immediately. The corresponding ExchOwningOrgMailResource07 instances are created at the same time.

- **Auto Allocation**

The auto allocation can be described as dynamic quota allocation.

When an organization subscribes an Exchange Service Plan as auto allocation (dynamicQuota = true), the capacity of resources the organization can consume is not determined. Exchange 2007 Resource manager only creates an ExchOwningOrgConsumer07 instance.

Now, only Business Organization can use auto allocation algorithm to consume Mailbox service. When a user in the organization tries to create a mailbox at the first time, the corresponding ExchOwningOrgMailResource07 instance is created, and then an ExchMailbox07 instance is created.

After the first time, when the organization tries to create a mailbox, the corresponding ExchOwningOrgMailResource07 instance is reallocated according to the size of new mailbox. If the older ExchMailStore07 cannot offer enough capacity, a new ExchOwningOrgMailResource07 will be created.

So, the mailstore quota of the organization applies auto allocation is dynamic.

### Resource Sharing

There are two types of resource sharing:

- If a resource is a “shared” mail store, it can be consumed by multiple organizations.
- If a resource is a “dedicated” mail store, it can only be consumed by one organization.

In current solution, only ExchStore07 resources can be used as “dedicated” resources. Exchange 2007 Resource Manager uses virtual capacity to differentiate “shared” and “dedicated” mail stores. Table 1 gives a full view of virtual capacity values in Exchange 2007 Resource Manager.

There is no API in BlockRMO that can change the virtual capacity of a resource without allocation. So, the ExchShareHogType07 instance is used as a common consumer to adjust VirtualFree properties of ExchMailStore07 instances by consuming virtual capacities. When Exchange 2007 Resource Manager adds a shared resource, the ExchShareHogType07 instance uses up all its virtual capacity. When a dedicated resource is added, the ExchShareHogType07 instance uses half of its capacity.

If an organization wants to consume “dedicated” mail stores, it must set “dedicated” parameter in its plan to “false”, and then Exchange 2007 Resource Manager will allocate available “dedicated” mail stores to it, according to [Mailstore Assignment Algorithms](#).

Table 1 Virtual Capacity usages in Exchange 2007 Resource Manager

Resource Type	VirtualMax	VirtualFree
ExchMailStore07 (shared)	3	0
ExchMailStore07 (dedicated)	2	1(before allocation)/0(after allocation)
ExchPublicStore07	1	1

### Mailstore Assignment Algorithms

There are 3 algorithms currently allowed in a Hosted Email 2007 Organization Plan to determine the ranking or ordering of the stores returned by a resource manager query for stores that meet the minimum requirements of the organization for the store, these are defined as follows:

- **tightPack**

The tightPack algorithm is to use the smallest number of mail stores.

- **level**

The level algorithm is to balance allocations evenly across available stores

- **minStoreCount**

The minStoreCount algorithm is to minimize the number of stores allocated to each organization, when algorithm levels Organization allocations across stores.

For tightPack, the list of available stores is returned in the order of increasing capacity. So, the store with the least capacity is chosen because it is first in the list.

For level the list of available stores is returned in the order of decreasing capacity So, The store with the greatest capacity is chosen because it is first in the list.

The minStoreCount algorithm is actually a combination of the two algorithms. When allocating a mailbox in an organization that already has allocation(s), the tightPack algorithm is used to determine the best store. However, when allocating a new organization or when allocating a new store for an organization the level algorithm is used.

This implementation could lead to some confusion, the first thing that you will notice is that there are really only two algorithms; second, you will notice that the behavior in these cases can be affected by a disparity in store size allocations. For example, if the level algorithm is used in an environment where a single store has more capacity than the rest, then all allocations will be made to that single store until the available space in that store is Equal to or Less than that of other stores.

## Typical Scenario: Life cycle of a Mailbox in RM

Table 2 shows a typical scenario of an Exchange mailbox's life cycle in RM. Every exchange RM related procedure will be elaborated in sequence.

Table 2

Step	Public procedures	Involved Exchange RM procedures
1	AddExchangeResources	AddExchangeResources
2	CreateOrganization	
3	Subscribe	AllocateOrganization
4	AddAvailablePlans	
5	CreateSMTPDomain	
6	CreateUser	
7	CreateMailbox	AllocateMailbox
8	DeleteMailbox	DeallocateMailbox
9	Unsubscribe	DeallocateOrganization

**Note:**

Procedure 1 and 10 are in Exchange 2007 Resource Manager namespaces. Procedure 2-9 are in Hosted E-mail 2007 namespaces.

## AddExchangeResource

### Description

AddExchangeResource could add 4 types of resources by calling:

- AddPublicStore\_: add public folder stores for business organization.
- AddMailboxStore\_: add mailbox stores for business organization.
- AddConsumerMailstore\_: add public folder stores for consumer organization.
- AddFrontEndServer\_: add front end server resource, but is never used.

The detailed process of each resource is similar. Therefore, only the AddMailboxStore\_ procedure will be elaborated as a sample.

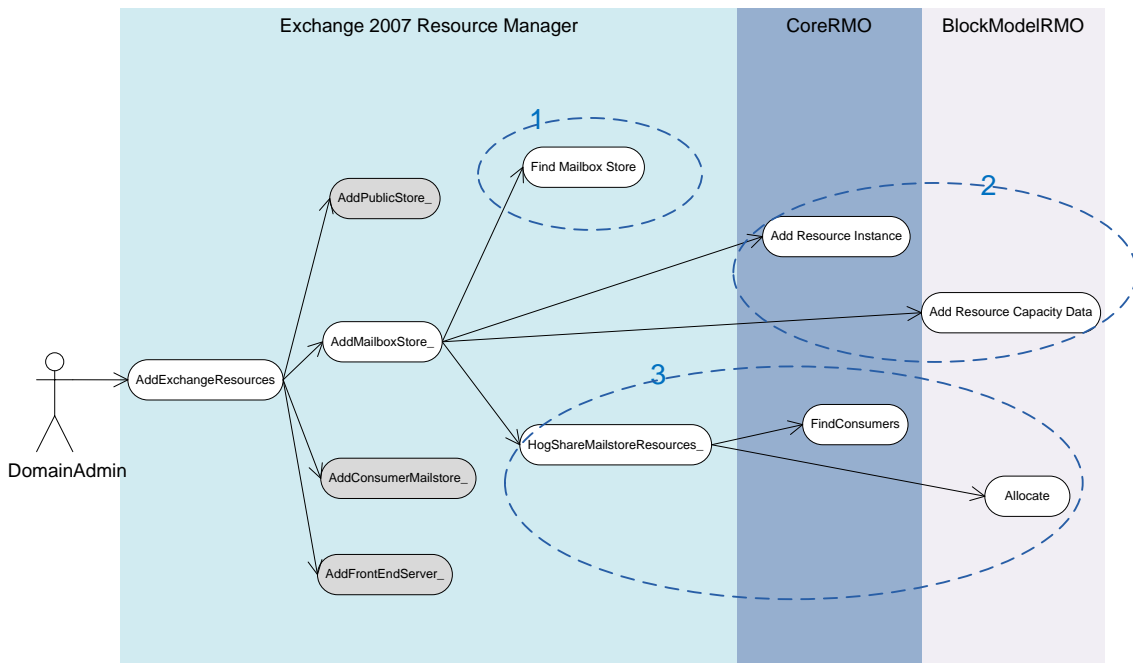


Figure 4 AddExchangeResources

1. Find the mailbox store for necessary information from Active Directory.
2. Add a new resource instance in [rmcore\_ResourceInstance] table, and add the mailbox store size into [blockmodel\_ResourceCapacity] table as the capacity data.
3. If the mailbox store is shared, add an allocation to consume the virtual capacity data in [blockmodel\_ModelAllocationData] table, indicating that the resource could be reused by any organization.

## Resource Manager Database reflection

Table 3

[rmcore\_ResourceInstance]: new resource instance of the mailbox store.

**Note:** The instance\_name field is the objectGUID of the mail store in Active Directory.

MP5SQL.Resou...urceInstance*								
instance_id	instance_name	instance_description	type_id	instance_location	instance_provis...	instance_online	instance_timest...	
f726d819-377c-...	8657E9F670A49...	Mailbox Database	0d2027e4-cb7d-...	ExMB	True	True	6/4/2008 11:26:...	

Table 4

[blockmodel\_ResourceCapacity]: resource capacity of the mailbox store resource instance.

MP5SQL.Resou...urceCapacity*								
instance_id	actual_max	virtual_max	actual_free	virtual_free	warning_pc	max_fill_pc	fill_ratio	
f726d819-377c-...	2000000	3	2000000	0	80	100	0	

Table 5

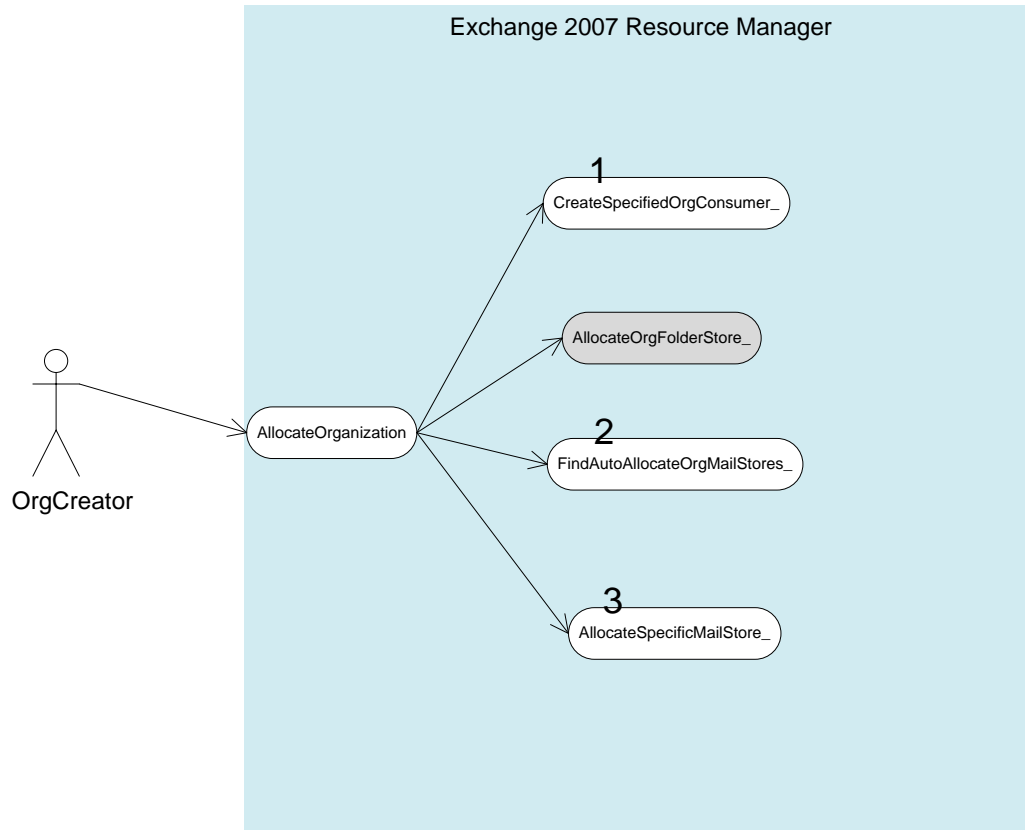
[blockmodel\_ModelAllocationData]: allocation of the shared resource.

MP5SQL.Resour...locationData*				
resource_id	consumer_id	actual_used	virtual_used	fill_ratio
f726d819-377c-...	d6cbd377-7982-...	0	3	0

## AllocateOrganization

### Description

AllocateOrganization allocates public folder and mailbox resources to business organizations. Only mailbox resource is elaborated here. Figure 5 shows a general logic for AllocationOrganization.



**Figure 5 AllocateOrganization**

1. Create organization consumer instance
  2. Find the mail store instance list
  3. Allocate the mail store to organization consumer
- Figure 6 is a detailed diagram of AllocateOrganization.

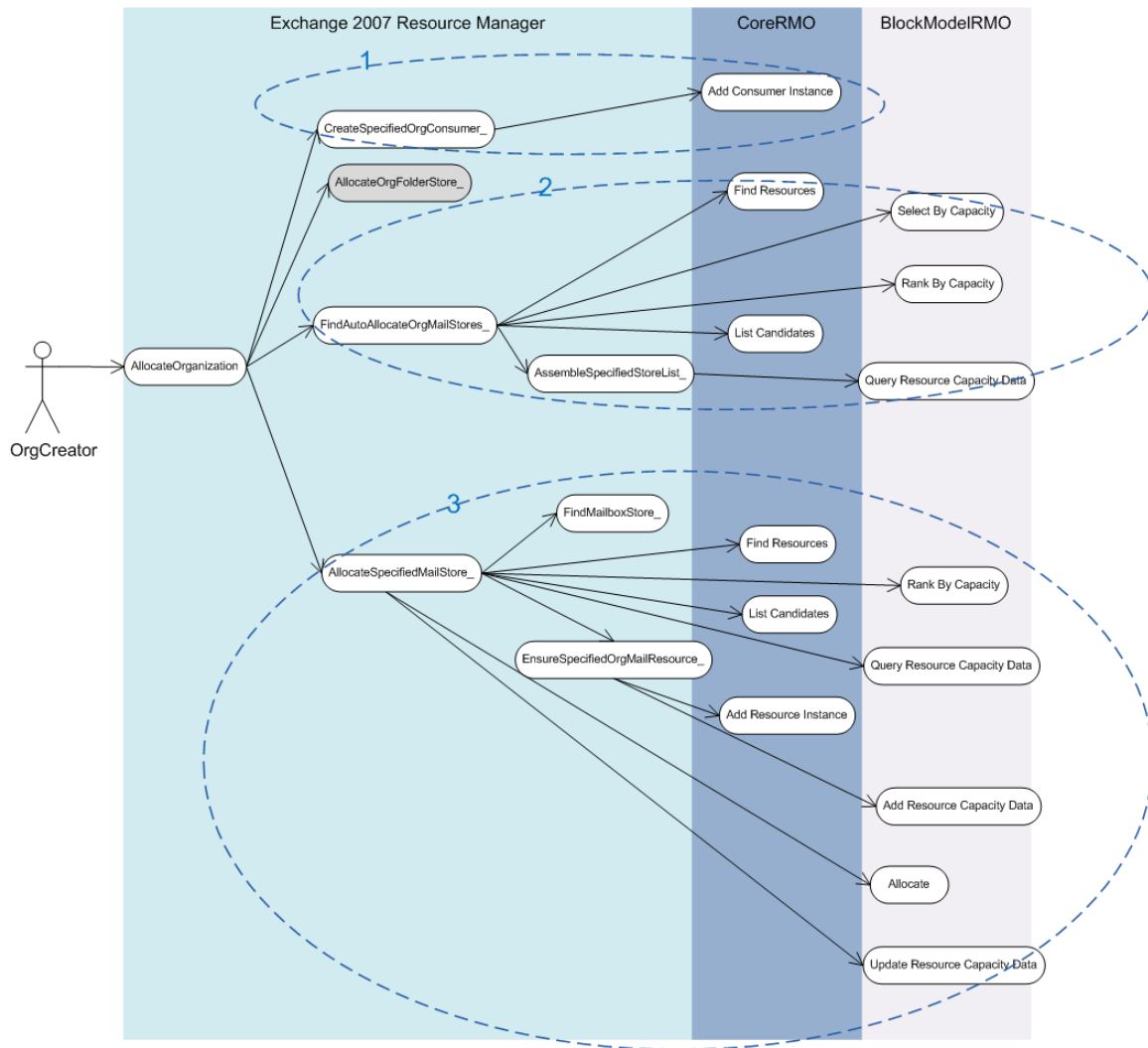


Figure 6 AllocateOrganization (detailed)

1. Create organization consumer instance  
This step is easy by using the AddConsumerInstance. A new consumer instance of the organization will be added into [rmcore\_ConsumerInstance] table.
2. Find mail store instance list  
Find all available mail stores, and assemble a list of mail stores which could satisfy the requested size.
3. Allocate the mail store to organization consumer  
This step expands the assembled list, and then makes allocation on each of them to meet the requested size. The allocation will be added into [blockmodel\_ModelAllocationData] table.  
Also, it will create an organization resource for the mailbox consumers within this organization. The resource instance is added into [rmcore\_ResourceInstance] table, and mailbox store size is added into [blockmodel\_ResourceCapacity] table.

## Resource Manager Database reflection

Table 6

[rmcore\_ConsumerInstance]: consumer instance of organization “AlpineSkiHouse”.

MP5SQL.Resou...umerInstance*				
instance_id	instance_name	instance_description	type_id	
49e5c0f5-6ede-...	22E801F8C3BD7...	Original path:LDAP://OU=AlpineSkiHouse,OU=ConsolidatedMessenger,OU=Hosting,DC=Fabrikam,DC=Com	25e30d18-dc4d-...	

Table 7

[blockmodel\_ModelAllocationData]: allocation of the mailbox resource to the organization.

MP5SQL.Resour...locationData*					
resource_id	consumer_id	actual_used	virtual_used	fill_ratio	
f726d819-377c-...	49e5c0f5-6ede-...	1000	0	0	

Table 8

[rmcore\_ResourceInstance]: new mailbox store resource instance of the organization. This could be consumed by the mailboxes under the organization.

**Note:** The instance\_name field is the objectGUID of the organization in Active Directory; instance\_location is the objectGUID of the mail store in Same as above.

MP5SQL.Resou...urceInstance*								
instance_id	instance_name	instance_description	type_id	instance_location	instance_provis...	instance_online	instance_timest...	
4599e6c5-eb3e-...	22E801F8C3BD7...	mail storage space owned by an organization	c8036f70-d95d-...	8657E9F670A49...	True	True	6/5/2008 12:28:...	

Table 9

[blockmodel\_ResourceCapacity]: capacity of the organization mailbox store resource

MP5SQL.Resou...urceCapacity*								
instance_id	actual_max	virtual_max	actual_free	virtual_free	warning_pc	max_fill_pc	fill_ratio	
4599e6c5-eb3e-...	1024000	0	1024000	0	80	100	0	

## AllocateMailbox

### Description

AllocateMailbox consumes the organization’s mailbox store resource, as shown in Figure 7.

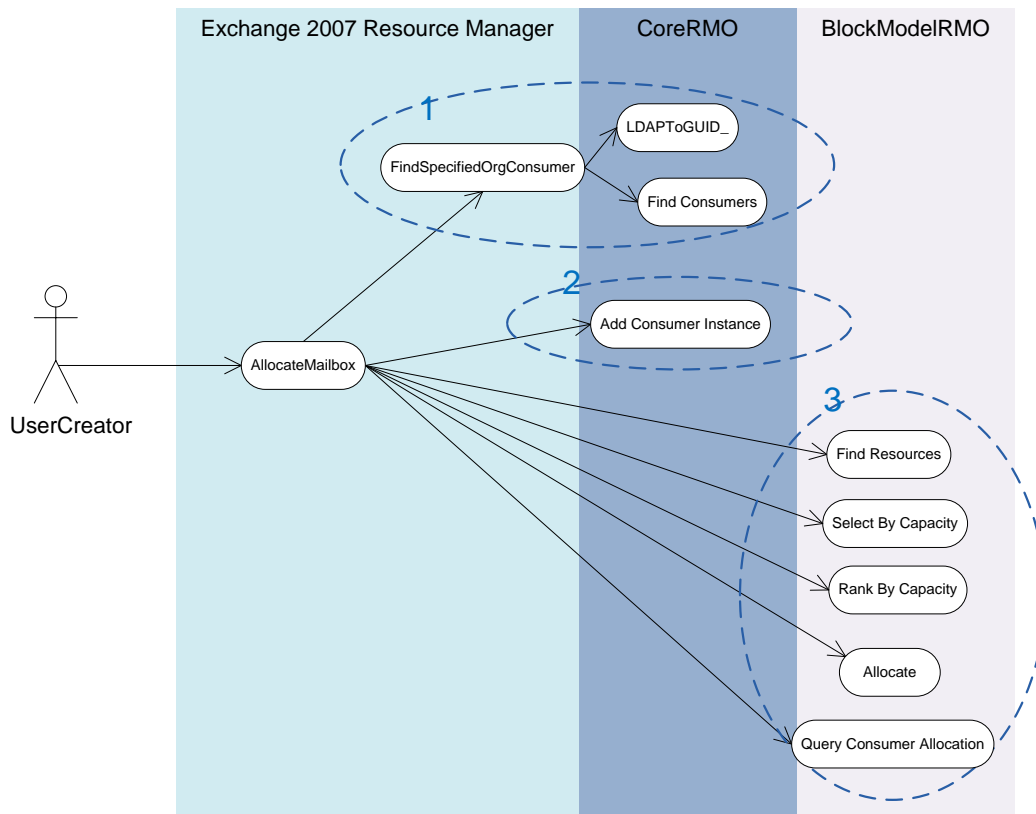


Figure 7 AllocateMailbox

1. Find the organization consumer instance, in order to find the organization's resource later.
2. Add a consumer instance of the mailbox into [rmcore\_ConsumerInstance] table.
3. Find the organization resource and allocate to the mailbox consumer instance into [blockmodel\_ModelAllocationData] table.

### Resource Manager Database reflection

Table 10

[rmcore\_ConsumerInstance]: consumer instance of the new mailbox.

MP55QL.Resou...umerInstance*			
instance_id	instance_name	instance_description	type_id
1270ad4d-cc51-4fd1-a44a-c...	E2DB1172FD77A94AAC0498DBD5CC8A55	mailbox	c76b4093-14a1-...

Table 11

[blockmodel\_ModelAllocationData]: resource allocation of the organization resource to the new mailbox consumer instance.

MP55QL.Resour...locationData*				
resource_id	consumer_id	actual_used	virtual_used	fill_ratio
4599e6c5-eb3e-...	1270ad4d-cc51-...	195313	0	0

## DeallocateMailbox

### Description

DeallocateMailbox is a mirror of AllocateMailbox.

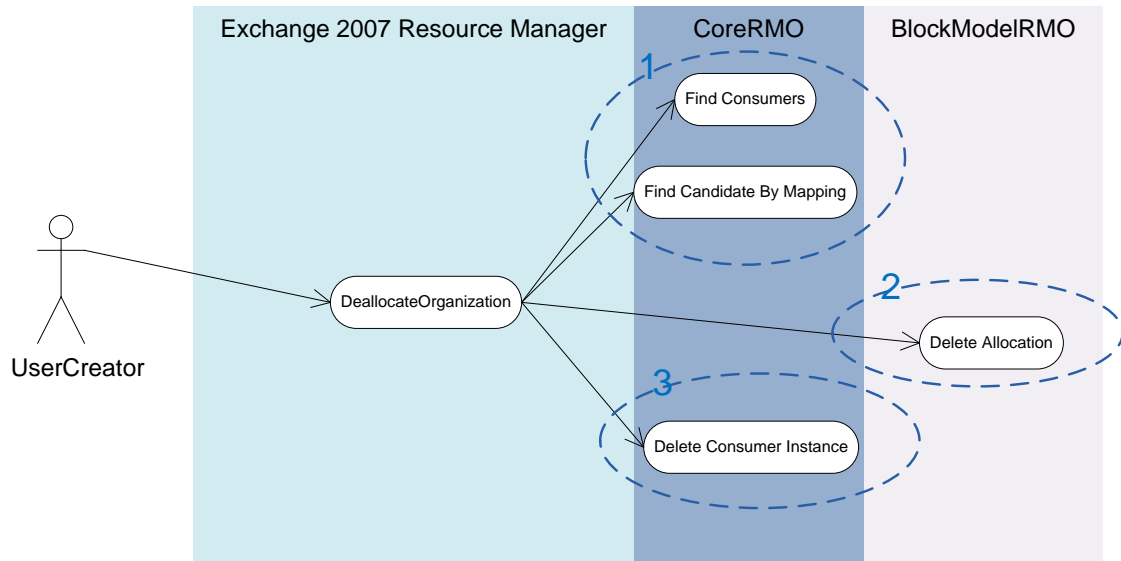


Figure 8 DeallocateMailbox

1. Find the consumer instance and the allocation of the mailbox.
2. Delete the mailbox allocation from [blockmodel\_ModelAllocationData] table.
3. Delete the consumer instance of the mailbox from [blockmodel\_ModelAllocationData] table.

### Resource Manager Database reflection

The consumer instance in Table 10 and the resource allocation in Table 11 are both deleted.

## DeallocateOrganization

### Description

DeallocationOrganization is a mirror of AllocateOrganization.

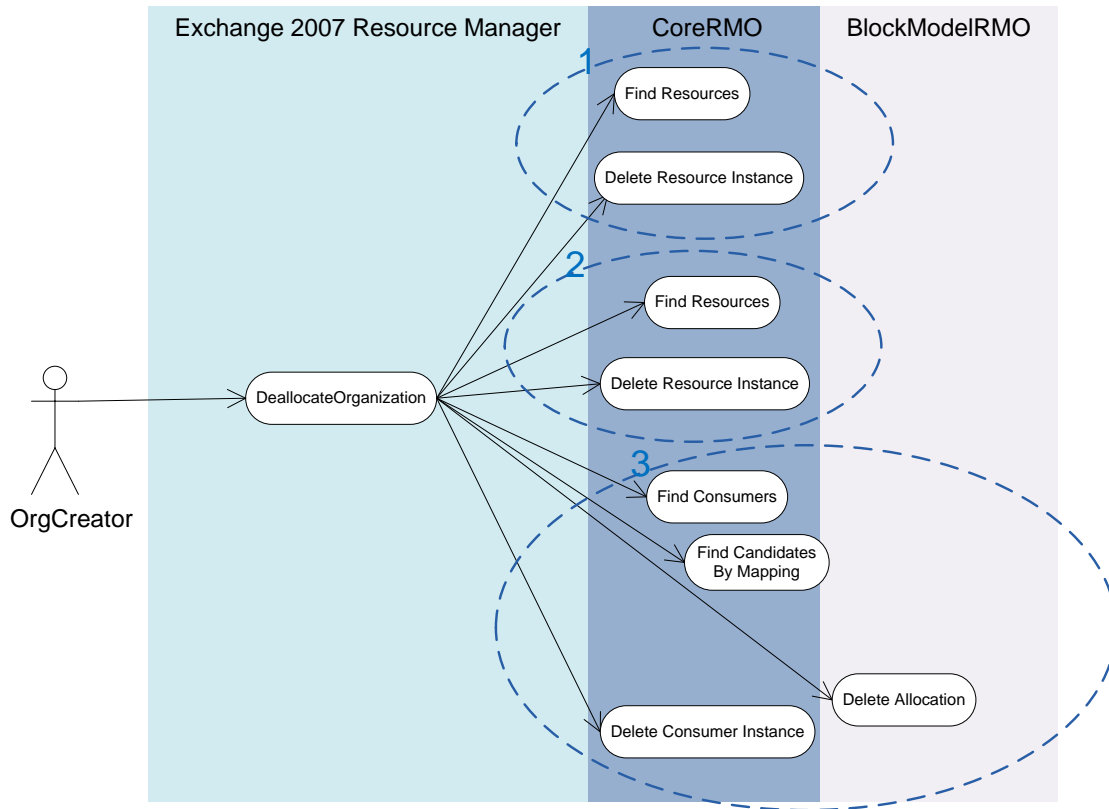


Figure 9 DeallocateOrganization

1. Find and remove the resource instance for public folder stores in [rmcore\_ResourceInstance] table.
2. Find and remove the resource instance for mailbox stores in [rmcore\_ResourceInstance] table.
3. Delete the allocation and consumer instance of this organization, in order to release the resources for reuse.

### Resource Manager Database reflection

The resource instance in Table 8 and resource capacity data in Table 9 are deleted.

Then the consumer instance in Table 6 and the resource allocation in Table 7 are deleted.

## RemoveExchangeResources

### Description

### RemoveExchangeResources is a mirror of **Note:**

Procedure 1 and 10 are in Exchange 2007 Resource Manager namespaces. Procedure 2-9 are in Hosted E-mail 2007 namespaces.

AddExchangeResource. It removes the public folder, mailbox store, consumer mailbox store and front server resource from being provisioned. Only mailbox store is elaborated .

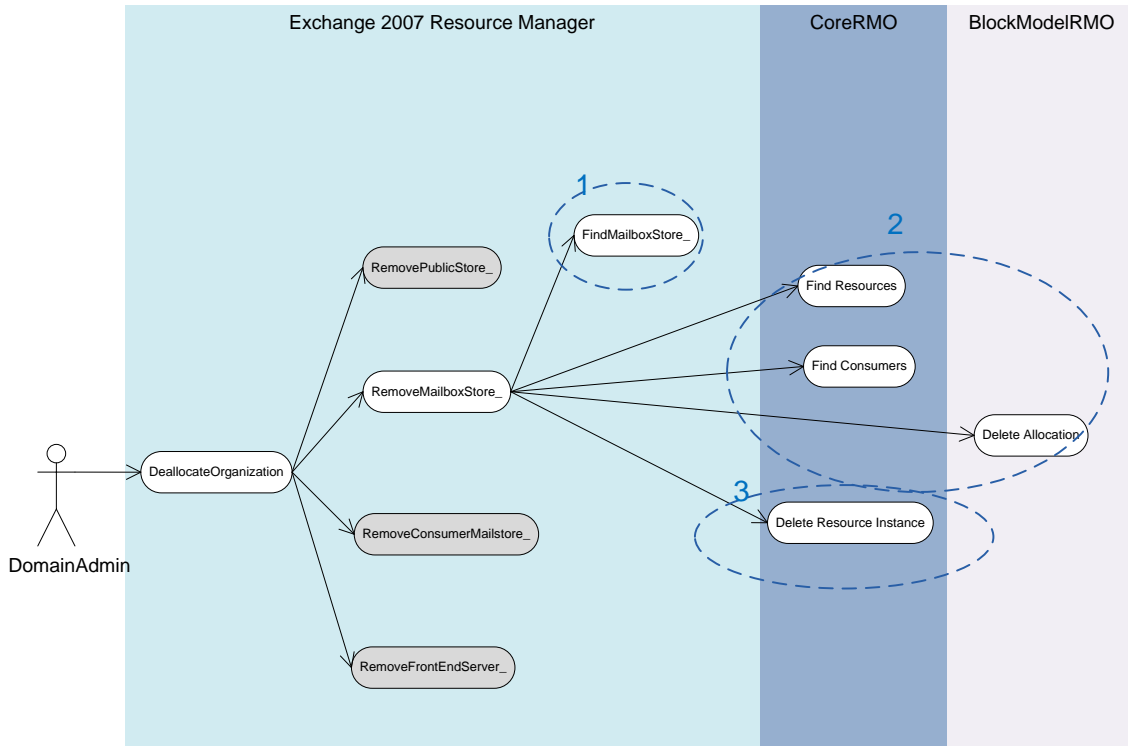


Figure 10 RemoveExchangeResources

1. Find the mailbox store and necessary information from AD.
2. Delete the allocation of shared resource in [blockmodel\_ModelAllocationData] table.
3. Delete the mailbox resource instance in [rmcore\_ResourceInstance] table.

### Resource Manager Database reflection

The allocation of shared resource in Table 5 is removed first.

Then the resource instance in Table 3 and resource capacity data in Table 4 are deleted as well.

## Glossaries

**CoreRMO** It is a core provider of MPF as part of Resource Manage Engine. It supports add, delete, query, and update operations for consumers and resources (and also their respective instances, types, and groups). It also establishes mapping relationships between consumers (instances and groups) and resources (instances and groups).

**BlockRMO** It is a core provider of MPF as part of Resource Manage Engine. It supports operations to define capacity for resource types and to manage allocations.