

CA BEST PRACTICES

CA AppLogic® Best Practices

Windows Appliance Creation Companion
and Example Guide v3 (June 2012)



LEGAL NOTICE

This publication is based on current information and resource allocations as of its date of publication and is subject to change or withdrawal by CA at any time without notice. The information in this publication could include typographical errors or technical inaccuracies. CA may make modifications to any CA product, software program, method or procedure described in this publication at any time without notice.

Any reference in this publication to non-CA products and non-CA websites are provided for convenience only and shall not serve as CA's endorsement of such products or websites. Your use of such products, websites, and any information regarding such products or any materials provided with such products or at such websites shall be at your own risk.

Notwithstanding anything in this publication to the contrary, this publication shall not (i) constitute product documentation or specifications under any existing or future written license agreement or services agreement relating to any CA software product, or be subject to any warranty set forth in any such written agreement; (ii) serve to affect the rights and/or obligations of CA or its licensees under any existing or future written license agreement or services agreement relating to any CA software product; or (iii) serve to amend any product documentation or specifications for any CA software product. The development, release and timing of any features or functionality described in this publication remain at CA's sole discretion.

The information in this publication is based upon CA's experiences with the referenced software products in a variety of development and customer environments. Past performance of the software products in such development and customer environments is not indicative of the future performance of such software products in identical, similar or different environments. CA does not warrant that the software products will operate as specifically set forth in this publication. CA will support only the referenced products in accordance with (i) the documentation and specifications provided with the referenced product, and (ii) CA's then-current maintenance and support policy for the referenced product.

Certain information in this publication may outline CA's general product direction. All information in this publication is for your informational purposes only and may not be incorporated into any contract. CA assumes no responsibility for the accuracy or completeness of the information. To the extent permitted by applicable law, CA provides this document "AS IS" without warranty of any kind, including, without limitation, any implied warranties of merchantability, fitness for a particular purpose, or non-infringement. In no event will CA be liable for any loss or damage, direct or indirect, from the use of this document, including, without limitation, lost profits, lost investment, business interruption, goodwill or lost data, even if CA is expressly advised of the possibility of such damages.

COPYRIGHT LICENSE AND NOTICE:

This publication may contain sample application programming code and/or language which illustrate programming techniques on various operating systems. Notwithstanding anything to the contrary contained in this publication, such sample code does not constitute licensed products or software under any CA license or services agreement. You may copy, modify and use this sample code for the purposes of performing the installation methods and routines described in this document. These samples have not been tested. CA does not make, and you may not rely on, any promise, express or implied, of reliability, serviceability or function of the sample code.

Copyright © 2012 CA. All rights reserved. All trademarks, trade names, service marks and logos referenced herein belong to their respective companies. Microsoft product screen shots reprinted with permission from Microsoft Corporation.

TITLE AND PUBLICATION DATE:

Windows Appliance Creation Companion and Examples Guide v3
Publication Date: June 28 2012

ACKNOWLEDGEMENTS

Principal Authors and Technical Editors

Michael Brennan
Gregory Buonaiuto
Roger Craig
Alexandre Moscoso
Terry Pisauro
Dhruv Shah
Lao-Tan Scotto
Anita Taucher

The principal authors and CA would like to thank the following contributors:

Joshua Shelton
Richard Simons

PRODUCT REFERENCES

This document references the following CA Technologies products:

- CA AppLogic
- CA Service Desk Manager

FEEDBACK

Please email us at impcdfedback@ca.com to share your feedback on this publication. Please include the title of this publication in the subject of your email response. For technical assistance with a CA Technologies product, please contact us at <http://ca.com/support>. For assistance with support specific to Japanese operating systems, please contact CA Technologies at <http://www.casupport.jp>.

DOCUMENTATION CHANGES

The following changes have been made since v2 was posted in February of 2012:

- Added information regarding AppLogic Enablement Kit on Cloud Commons
- Added notes regarding install size recommendations for Microsoft Windows operating system
- Clarified English language version of Microsoft Windows operating system was used for all test and as the basis for all steps provided
- Added reference to Windows Hostname Update field developed utility
- Updates to the Microsoft License Injection Key utility
- Clarified some steps based on feedback
- Added information on enabling installation of VMware tools for appliances created on a non-ESX grid but later ported to an ESX grid

Contents

Chapter 1: Introduction	11
What this Guide Provides.....	11
Who should read this guide?	12
Where to go for more information	13
Product Documentation, Education and User Forums	13
Technical Support.....	13
CA AppLogic Best Practices	14
Cloud Commons Resources	14
Chapter 2: Appliance Overview and Design Guidelines	15
Appliances and Applications	15
Getting Started.....	16
But first, a Word about Licensing	16
Identify Configuration Details	17
Resources	18
Volumes.....	19
Terminals.....	20
Appliance vs. Application Boundaries	21
Project Design Considerations	22
Naming Conventions	22
Boundary Property Tips.....	25
Terminal Tips.....	25
Volume Tips.....	26
Script Tips and Directory Structure	26
Standardize Default Passwords	27
Use Annotations on your Design Canvas.....	28
Add Notes when Updating Boundaries	28
Chapter 3: Creating a Windows 2008 R2 Appliance	29
Before you Begin.....	29
Preparing the Grid.....	29
Preparing the Grid Hardware to run Windows.....	29
Import the Filer and system_ms Catalog.....	30
Creating a Windows 2008 Appliance	32
Step 1: Create the WIN08 Singleton.....	32
What Happens If I Receive a Permission Denied Message?	36
Step 2: Configure and Update Windows	37
Step 3: Activate Windows.....	42
Step 4: Post Install Configuration	42
Step 5: Configure the Singleton to be Fully Managed by CA AppLogic.....	44
Step 6: Optional Configurations	48
Step 7: Modify the Boundary and Instance Settings	51
Step 8: Prepare the Singleton to be Added to a Catalog	53
Step 9: Add the Singleton to a Catalog.....	58
Step 10: Verify that the Appliance Starts	59

Creating a Windows 2008 Filer Appliance	60
Step 1: Create a Windows Server 2008 32-bit Appliance.....	61
Step 2: Configure Windows	63
Step 3: Set the Singleton to be Fully Managed	66
Step 4: Complete Windows Configuration.....	68
Step 5: Modify Boundary and Instance Settings	69
Step 6: Add the Singleton to a Catalog	70
Step 7: Create a Windows Server 2008 Filer Appliance	70
Step 8: Verify the Filer	72
Chapter 4: Creating a Microsoft SQL Server Appliance	75
Step 1: Create an AppLogic Application	75
Step 2: Modify WIN0864S to Install MS SQL Server.....	76
Step 3: Prepare Windows to Install MS SQL Server	79
Can I change the node name after MS SQL Server is installed?.....	80
What if there are multiple MS SQL Server appliances in the same application?	80
Step 4: Install MS SQL Server	81
Step 5: Install the MS SQL Server MSI.....	82
Verify that the Paging File was Re-Created Correctly with a New Volume	83
What if I See a Message Saying a Temporary Paging File was Created?.....	85
Step 6: Script the Operations to License MS SQL Server Upon Startup	85
Step 7: Prepare the MS SQL Server Singleton to be Added to a Catalog	86
Step 8: Add the Singleton to a Catalog	88
Step 9: Verify your MS SQL Server 2008 Appliance	88
Chapter 5: Creating a Microsoft IIS Appliance	91
Step1: Create the AppLogic Application to Install IIS.....	91
Step 2: Install IIS	92
Step 3: Install the IIS MSI	93
Step 4: Prepare the Singleton to be Added to a Catalog	94
Step 5: Add the Singleton to a Catalog	98
Step 6: Verify the IIS Appliance.....	98
Chapter 6: Application Overview and Design Guidelines	101
Application Design Considerations	101
Application Properties vs. Appliance Properties.....	101
Singletons, Appliances and Data Persistence	102
Application User Volumes vs. Appliance Volumes.....	103
The Importance of Appliance Start Order.....	103
Communications between the Application and the Network	105
Networking setup	106
Chapter 7: Creating a Microsoft IIS Application	111
Step 1: Build a new CA AppLogic Application	111
Step 2: Add Appliances to the New Application	112
Step 3: Connect the Appliances	113
Step 4: Add Application Properties.....	115
Step 5: Create Application User Volumes	116

Step 6: Configure the Instance Settings of Each Appliance	117
IN Gateway Appliance	117
IIS0864S Appliance	118
SQL Appliance.....	119
NET Gateway Appliance	120
MON Appliance	121
Step 7: Configure the Application	122
Step 8: Test the Application	123

Chapter 8: Creating a CA Service Desk Application **125**

Step 1: Create the Application	126
Step 2: Branch the Classes	127
Step 3: Resize the Disks.....	128
Troubleshooting	128
Step 4: Modify Class Boundaries	129
IN Singleton	129
WIN0864S Singleton.....	130
SQL0864S Singleton.....	132
Step 5: Add Properties to the Application Boundary	134
Step 6: Add the Application User Volumes for the MON Appliance	135
Import ISO Images.....	135
Add an Application User Volume for the MON Appliance.....	136
Step 7: Modify Instance Settings.....	136
SDM_IN_12_6	136
SDM_PRI_12_6_S.....	137
SDM_MDB_12_6_S	138
NET	140
MON	141
Step 8: Configure the Application	141
Step 9: Start the Application	142
Step 10: Login to the Singletons.....	143
Login through Remote Desktop.....	143
Graphical Console.....	143
Step 11: Prepare Windows to install CA Service Desk Manager r12.6	143
Verify that the Volumes were Created.....	143
Verify the Hostname has been Changed	144
Step 12: Install the MDB.....	145
Step 13: Install the Microsoft SQL Server Client.....	145
Step 14: Install CA Service Desk Manager	146
Step 15: Create the Scripts.....	148
Microsoft SQL Server Client Script.....	148
CA Service Desk Manager Script.....	148
Step 16: Prepare the Singletons to Become Appliances	151
Step 17: Store Appliances in the CA AppLogic Catalog.....	152
Step 18: Test the Application and Finalize the Application	152

Chapter 9: Advanced Application Configuration Tips	155
Application Templates	155
Create an Application Template	156
Provision and Deploying an Application from an Application Template.....	156
Assemblies	157
Designing Assemblies.....	158
Step 1: Installing the Application	158
Step 2: Configure the Assembly Class	161
Step 3: Property Management.....	162
Step 4: Testing the Application	166
Step 5: Save the Assembly	167
Using the Assembly.....	167
Create an Application with an External SQL Server	167
Maximizing Appliance Re-usability	169
Appendix A: Using the Microsoft Product Key Injection Utility Licensing Scripts	171
Supported Microsoft Products.....	171
Before You Begin	171
Microsoft Product Installation	172
Microsoft Windows Server 2008 R2	172
Microsoft SQL Server 2008 R2	173
Example Script Commands for Licensing	174
Microsoft Windows Server	174
Microsoft SQL Server 2008 R2	174
Implementation	175
Step 1: Install the License Key Injection Utility Script	175
Step 2: Modify Group Policy	176
Script Execution	179
setProdKeyWin2008r2.ps1 Script Properties.....	180
License Removal	185
Troubleshooting Tips	185
Appendix B: Microsoft SQL Server 2008 R2 64-bit Configuration File	193
Configuration File Contents	193

Chapter 1: Introduction

CA AppLogic® software, a turnkey application-centric cloud platform, enables you to package comprehensive applications into portable self-contained entities giving you the ability to more quickly create and provide new virtualized business services. In CA AppLogic, applications are comprised of appliances which also include the underlying infrastructure, such as the operating systems, firewalls and network communications. The CA AppLogic catalog includes several base appliances that you can use as-is or modify to build your own application. You can also create your own appliances from scratch if you prefer – or if the existing base appliances are not applicable for your use.

Information on how to create and modify appliances, as well as how to use them to build applications, can be found in the CA AppLogic product documentation. This document provides examples to further demonstrate those processes, walking through the step-by-step creation of Microsoft Windows, SQL Server and IIS appliances, as well as IIS and CA Service Desk Manager applications. It also provides additional appliance design guidelines based on the experiences of the Appliance Development team.

What this Guide Provides

Each chapter in this guide is designed to build upon the concepts and examples provided in the previous chapter. Therefore it is highly recommended that you read them in the order in which they are presented.

Topics include the following:

- **Chapter 2: AppLogic Appliance Overview and Design Guidelines**

This chapter introduces the basic concepts behind creation of an appliance and provides design guidelines and gotchas.

- **Chapter 3: Creating a Windows 2008 R2 appliance**

This chapter applies the basic concepts introduced in the previous chapter to create a sample Windows 2008 r2 appliance as well as a Windows 2008 Filer appliance.

- **Chapter 4: Creating a Microsoft SQL Server Appliance**

This chapter demonstrates how build a Microsoft SQL Server appliance, which will be used in conjunction with the Microsoft Windows 2008 R2 appliance in later chapters.

- **Chapter 5: Creating an IIS Appliance**

This chapter demonstrates how to utilize the concepts learned in the previous two chapters to build a web service appliance.

- **Chapter 6: Application Overview and Design Guidelines**

This chapter provides an overview of the application creation process and design considerations.

- **Chapter 7: Creating an IIS AppLogic Application**

This chapter demonstrates how to link the example appliances together to build a simple web server application.

- **Chapter 8: Creating a CA Service Desk Manager Application**

This chapter demonstrates how to construct a more complex two-tier CA Service Desk Manager application.

- **Chapter 9: Advanced Application Configuration Tips**

This chapter explores additional application configuration tips, including design for re-usability, use of assemblies (also known as “composite appliances”) and creating an application which uses in external SQL server.

- **Appendix A – Using the Microsoft Product Key Injection Utility**

This appendix provides details on how to use the field developed Microsoft Product Key Injection Utility to manage licensing for Microsoft Windows 2008 r2 and Microsoft SQL Server 2008 r2.

- **Appendix B – Microsoft SQL Server 2008 R2 64 bit Configuration file**

This appendix includes the contents of the configuration script that used to silently install Microsoft SQL Server 2008 R2 in the examples.

Note that the steps and references included in this document are based on CA AppLogic r3.0, however, however much of the information is still applicable for later releases.

Who should read this guide?

This guide is intended to be read by software developers and application architects who want to create new appliances and applications that can be deployed for use on a CA AppLogic grid environment. Grid maintainer access is required to perform certain steps documented in this guide, and is clearly noted where applicable. See the CA AppLogic product documentation for a more complete discussion of typical user roles and responsibilities.

Although an overview of the basic concepts and design considerations is provided in the next chapter, familiarity with the existing CA AppLogic documentation is assumed, as is access to a functioning CA AppLogic grid system.

Where to go for more information

For more details on creating applications and appliances in CA AppLogic, as well as any other topics discussed in this guide, there are many resources you can consult.

Product Documentation, Education and User Forums

The documentation bookshelf for CA AppLogic r3.0 can be accessed through the following link:

<https://support.ca.com/cadocs/0/CA%203Tera%20Applogic%20%209-ENU/Bookshelf.html>

In addition product training is available from the CA Learning Site:

https://calearning.ca.com/plateau/user/cadeeplink.do?linkId=CATALOG_SEARCH_RESULTS&siteID=United+States&keywords=applogic

Note: Although some of the courses listed may be written for earlier CA AppLogic releases, much of the material is still applicable (unless otherwise noted).

Finally, there are also interactive forums where users can post questions, request information, or offer insight and suggestions. Available forums include:

<https://communities.ca.com/web/ca-3tera-global-user-community/welcome-3tera-forum-users>

To access the message boards and forums you will need to first create a CA Support Online account profile - if you do not already have one - and login in to Support Online. If you previously participated in the 3Tera AppLogic Forums, your information has been transferred to the new CA 3Tera Global User Community - and your user name will be changed to your CA user ID associated with your email address. If you are a new user to the community you need to join the community first (click "Join this Community" link).

Technical Support

Technical support for CA AppLogic is available through the following link

<http://support.ca.com>

The product home page for CA AppLogic is

<https://support.ca.com/irj/portal/prddtlshome?productID=8383>

CA AppLogic Best Practices

Additional best practice tips and guidelines are available through the following link:

https://support.ca.com/phpdocs/0/common/impcd/r11/Cloud/AppLogic_Frame_sc.htm

The tips provided through this link are based on the in-house experience of CA Technologies developers and support staff and are supported on an as-is basis.

Cloud Commons Resources

Cloud Commons (<http://cloudcommons.com>) provides a rich source of information about working with the CA AppLogic platform – as well as broader topics related to the cloud environment. In addition to the articles and blog entries available through the Learn tab, there are user groups that can be accessed through the Collaborate tab:

<http://cloudcommons.com/web/cc/collaborate>

Scroll through the list and click the respective links to join or request access to “Open” or “Members Only” user groups that interest you – or create your own group. To view the AppLogic Enablement Kit request to join the AppLogic Partners user group. The AppLogic Enablement Kit provides a roadmap for locating the information you need to get up and running quickly with CA AppLogic. It also includes helpful video snippets and a forum for accessing additional information.

Cloud Commons also hosts Developer’s Studio – accessed through the Build tab. Joining Developer’s Studio enables you to host your project using the Cloud Commons Work Bench (based on FusionForge), and provides access to tools to help your team collaborate and manage key aspects of your project, including:

- Managing tasks, documents, and file releases
- Creating and controlling access to source code
- Taking part in surveys about best practices (and gaining access to the results of those surveys)
- Tracking issues
- Testing the final project – by reserving time on the Test Grid.

Cloud Commons is also home to the Cloud Commons Marketplace where applications, appliances and other content can be offered for sale and downloaded for purchase. Additional information on using the Marketplace can be found in the Cloud Commons Help Center link.

Chapter 2: Appliance Overview and Design Guidelines

This chapter provides a basic review of what goes into planning and building a CA AppLogic appliance. It is not intended as an introduction to CA AppLogic – for that you should consult the product documentation.

Also included are tips and best practices for managing your appliance project. The guidelines and best practices outlined here will be used in the examples in the rest of the guide.

Appliances and Applications

Depending on your business requirements, you may be tasked with delivering just the individual appliances which clients can drag from the CA AppLogic catalog, drop onto the design canvas and use to build their own custom configurations, or with development of ready-to-provision applications or application templates – or both.

To understand the differences between these options, let us first review some terms:

- An **appliance** is a generic term used for individual virtual machines that emulate physical hardware like servers, switches, routers and load balancers and it represents the basic building block for more complex appliances and applications. In CA AppLogic, however, an appliance is more than just a virtual machine. Each CA AppLogic appliance executes in its own virtualized environment, boots its own operating system, application services and other required software. It also has a **boundary** which isolates the interior of the appliance from the exterior – defining both the structure of the appliance and how it interacts with other CA AppLogic components – including the grid.
- Appliance **classes** are re-usable device definitions whose boundary attributes can be modified to adapt the device's behavior for a specific purpose. For example, after dragging a web server appliance class from the catalog onto the canvas, you can create a **singleton instance** of that web server class and modify the boundary properties to suit your particular environment. Those changes can even be saved as a new class.
- Both appliance and assembly classes are stored in a CA AppLogic **catalog** where they can be dragged and dropped onto the AppLogic editor canvas to create new instances. When you create a new appliance you can add the class to the **local catalog** where it can be used by the application or add it to the **global** catalog, where it can be accessible to other applications and users.

- Appliances and assemblies can be grouped together as **applications** which represent a single system object that includes everything necessary to provide a specific service in a distributed architecture, such as a CRM system, a PBX or an e-commerce deployment. In addition to software implementation binaries an AppLogic application will include all html pages, templates and scripts, databases and content, firewalls and all configuration information needed to reconstruct and run the application on the CA AppLogic grid. As with appliances and assemblies, CA AppLogic applications have boundaries through which attributes can be manipulated to manage interaction between the application and the “outside world”.
- **Application templates** include all pieces required to establish a working instance of the solution – including the necessary database and input/output gateway appliances *preconfigured for the specific architecture*. They are designed to be provisioned to quickly stand up a working instance of the solution.
- Communication between appliances *within* an application is managed through input and output **terminals**.
- Communication between applications and the “outside world” are managed through specialized appliances called **gateways**. Gateway devices serve as relay agents to pass communications to/from the external (public) network to devices on the boundary interior connected to appropriate terminals on their boundaries. You can opt to use the default gateway appliances that are included with the standard catalog, such as IN and NET, or create a custom gateway to handle communications specific to your appliance.

Following are some additional tips for creating appliances. For further guidelines pertaining to applications see [Chapter 6: Application Overview and Design Guidelines](#).

Getting Started

CA AppLogic includes several appliances - as well as two generic appliance templates – that you can use as a starting point for building your new appliance. You can also purchase prebuilt appliances through other sources such as Cloud Commons (www.cloudcommons.com). To create a new appliance class from an existing class you branch that class. This creates a new singleton class. You can then add additional functionality to that existing class, such as an upgrade or further parameterization.

Note: CA AppLogic also enables you to create a new class from scratch using the Appliance Productization Kit, however, that option is beyond the scope of this document. Refer to the *CA AppLogic Appliance Productization Kit User Manual* for more details.

But first, a Word about Licensing

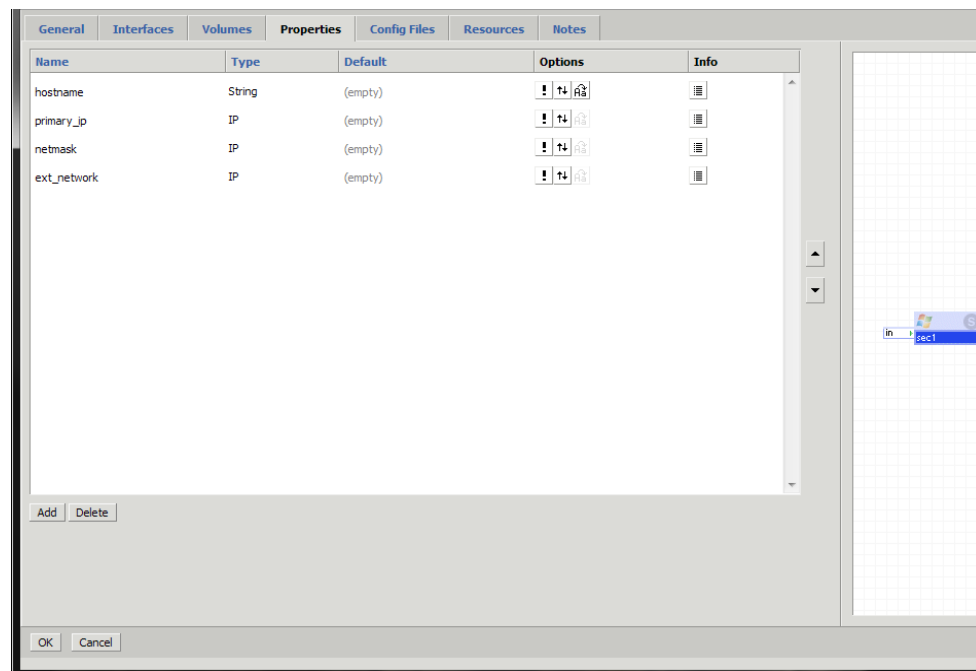
Identifying which software applications you will be including with your appliance, as well the licensing requirements and re-distribution restrictions that apply to those applications, is a critical part of the appliance planning process. The consequences of failing to adhere to those requirements can be prohibitive.

Since we are installing several Microsoft products in our examples we have elected to create three new boundary properties – kms_server_port, win2008r2_prod_key and sql2008r2_prod_key – which the end user is required to configure upon startup of the appliance. These values are then used by custom startup scripts to retrieve and inject the valid licensing information into the product for activation. This approach shifts responsibility for obtaining the valid license to the end user and is part of the field developed Microsoft License Key Injection utility (see [Appendix A](#) for more details). Depending on your particular licensing requirements and how you plan to use the appliance in your environment, you may choose to create your own configuration scripts or follow a different approach altogether.

Identify Configuration Details

User defined configuration information - such as hostname IP addresses, gateways and DNS server name – is passed between appliances, assemblies, and applications through their **boundaries** which act as an implicit firewall for these components.

Here you can see an example of boundary properties tab in the class editor:



Once the property details are configured the values are stored in the `/var/run/applogic/appliance.conf` file. For example:

```

root@Grid4:~
hostname=sec1
primary_ip=172.16.1.4
netmask=255.255.255.0
ext_network=141.202.0.0
_COMP_CLASS=.sec1
_COMP_NAME=main.sec1
_APP_NAME=BP_SDM-VPN1
~
~
~
~
~
"appliance.conf" 7L, 143C      1, 1      All

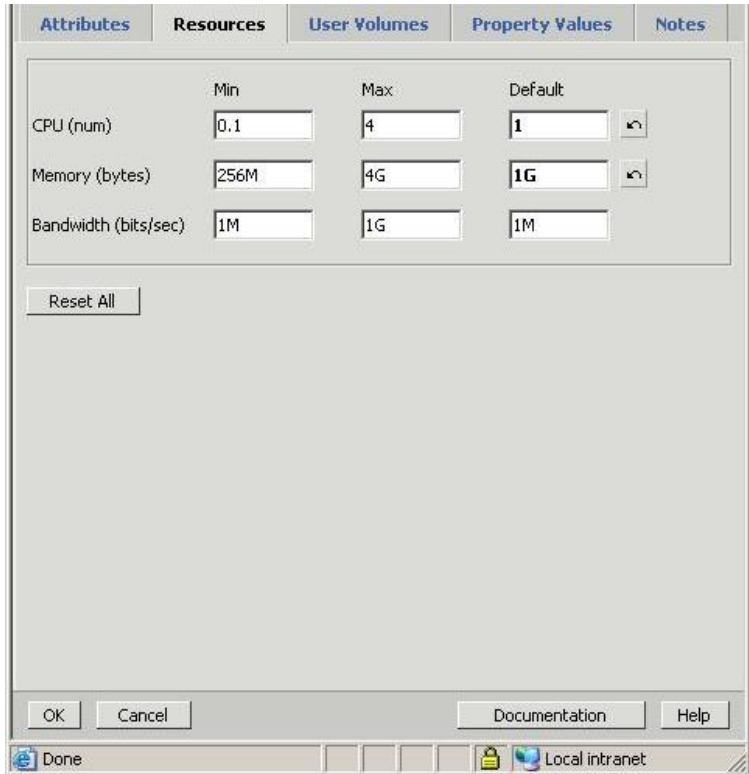
```

In determining which boundary properties to include think about those items in all of the various components of the appliance that must be variable. Common examples include network details, such as IP addresses and port numbers, and service names. Those items must be made part of the boundary – and can be either mandatory or optional. Be sure to expose all configuration parameters that need to be set during the appliance setup.

Boundary definitions are stored as descriptor files that are packaged with the binary image of the virtual environment that is your appliance. They can be created or modified through the Class Editor – which is part of the CA AppLogic Application Editor.

Resources

Resources identify the minimum, maximum and default amount of CPU, memory and bandwidth required by the appliance. Here you can see an example of the Resource tab contents:

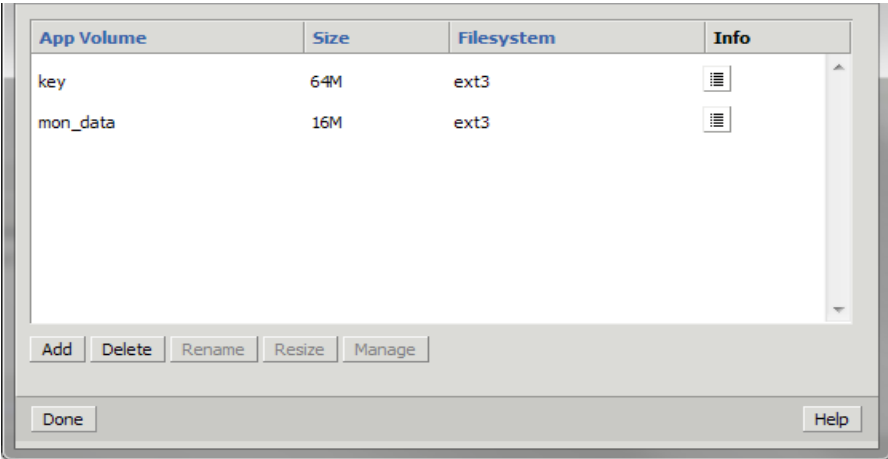


CA AppLogic automatically calculates the resource range of an application based on the resource range of all of the appliances used in the application. You can leave these values as-is or add further constraints – either by setting a resource range or designating a subset of servers in the system on which the application can be scheduled.

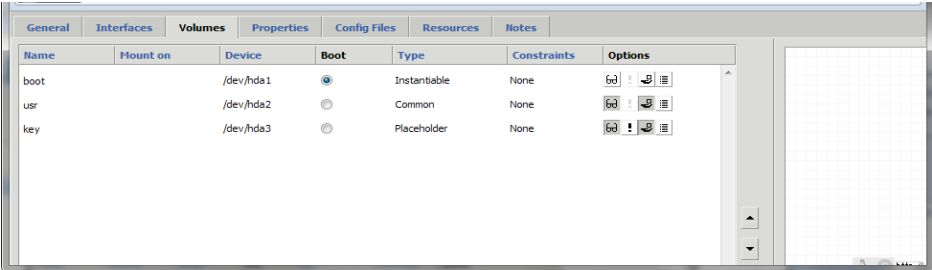
Volumes

Each appliance needs at least one storage volume – the one from which it boots. These volumes are provided as part of the class definition of the appliance and instantiated whenever an instance of the appliance is created.

Here you can see an example of the Manage Volumes screen in the Application Editor:



And here an example of adding a volume to an appliance boundary:



Tip! Including a placeholder for additional volumes when you first define an appliance will enable you to more easily plug-in an external volume later on. Volumes “plugged” into a placeholder slot are known as “application volumes.”

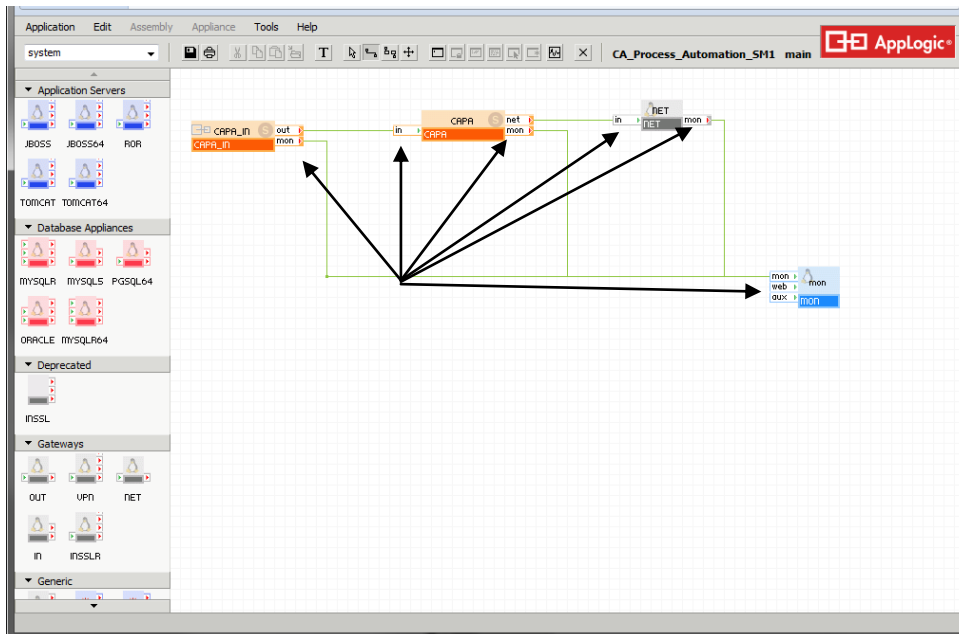
Be mindful of the size of the data volumes you create. Consider separating data into specific volumes or shared data sources. This can be done by creating a symbolic link from a configuration directory to a 'config' volume mounted as an application level volume for the app.

Note: In addition to licensing requirements, compliance with third party software redistribution requirements may dictate that separate volumes be used to segregate the third party software from your solution software.

Terminals

Terminals are connection points for logical interactions between appliances – designed so that existing software packages inside virtual appliances can communicate through terminals without requiring modification. Looking from inside an appliance, the terminal is a host name visible only to that appliance instance. The terminal name of an input terminal can be used inside the appliance to set up a listening socket for accepting connections. The terminal name of an output terminal resolves to whatever appliance is connected to the output and can be used to establish connections to that appliance.

Here you can see examples of terminals and how they are connected:



Terminals can be used for:

- Input – for accepting network connections
- Output – for originating network connections

Both types of terminals are bi-directional – requests and data can flow both into and out. A terminal consists of a network name, virtual network adapter and virtual network interface.

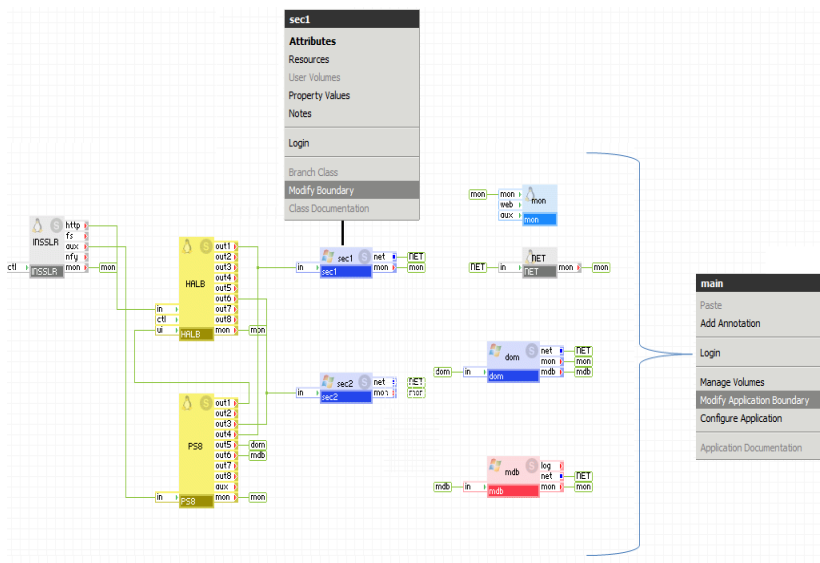
When an output of one appliance is connected to an input of another appliance, CA AppLogic creates a virtual wire between their respective virtual network interfaces and assigns virtual IP addresses to both ends of the connection.

For example, if you are building an appliance that requires access to a database server you can define an output for accessing that database and call it “DBASE”. When configuring the JDBC driver inside the appliance you only need to set the name of the output “DBASE” as the host name of the target database server. At run time, each instance of the appliance may be connected to a different database server without having to change the target host name for the JDBC driver – the same host name “DBASE” will automatically resolve to the correct IP address of the database server to which the particular instance is connected.

For each terminal you need to specify what protocol(s) is allowed on it – creating, in effect, a virtual firewall on the terminal. Any terminal can be marked as mandatory – if a required terminal is not connected CA AppLogic will not start the application.

Appliance vs. Application Boundaries

Boundaries can be configured on multiple levels. For example, here you can see an example of both application and appliance level boundary definitions:



When property values are defined at the *appliance level*, they become transparent to the appliance’s boot environment. You can leverage this by using scripts to export the property values as environment variables or even pass them to external scripts that can perform any number of configuration tasks.

When property values are defined at the *application level* they can be inherited by individual appliances, which can then make this property value available to the appliance level environment. This inheritance model allows you to set application level values that can be used to configure the entire application. By setting values for all mandatory values, an application with properly built appliances (this includes the scripts required to configure the application), essentially configures itself by utilizing the inheritance model to the underlying appliances within an application.

Tip! The `allowed_hosts` and `denied_hosts` boundary properties each designate an IP mask that can be used as a filter to either allow or deny passage of traffic from the host IP address(es) matching the specified pattern. The default value for `allowed_hosts` is "0.0.0.0/0" which allows traffic from all hosts. The default value for the `denied_hosts` property is blank – meaning that no host traffic is explicitly denied. These properties are typically exposed on the input appliance (either a custom IN gateway or the standard CA AppLogic IN gateway) however; if you are creating an application template as well the best practice recommendation is to expose them there, too, to ensure that security is properly set upon provisioning.

Project Design Considerations

Establishing a clear and consistent set of standards that will be adhered when you are building an AppLogic appliance or application enables you to more efficiently make use of the resources at hand and to more effectively leverage the resulting components for future updates and additions. Examples include:

- naming conventions
- default password configuration
- standard file locations

Naming Conventions

Naming conventions should be established for the following categories of items:

- Appliances, assemblies, applications and application templates
- Boundary items, including terminals and properties
- Catalogs and categories
- Supporting documents and scripts

Following are some tips for developing naming conventions for these items.

Naming Appliances, Assemblies and Applications

In choosing a class name for your appliance, assembly or application, consider the following:

- Use the key package's name only if that knowledge is critical

- Use all uppercase class names
- Where possible, limit the name to a maximum of 5 to 6 characters.

Note: Longer names are truncated in the shape (class field).

Keep an extra position open as the editor appends an instance number to the class name for each instance.

For example, we will be using the following naming convention for the Microsoft products in this guide:

PRDxyz

Where:

- *PRD* refers to the product that serves the primary function. For example **WIN** for Microsoft Windows Server, **SQL** for Microsoft SQL Server, and **IIS** for Internet Information Services.
- *x* denotes the version – either of the product itself, or the operating system version. For example, **08** would refer to the 2008 version of the Windows operating system while **03** would refer to the 2003 version.
- *y* refers to the platform. If the platform is not noted in the name, it is assumed that it is a 32 bit appliance; otherwise **64** would be specified to signify that it is a 64-bit appliance.
- *z* denotes a specific edition either of the product itself, or the operating system edition. For example, **S** refers to Standard edition, **E** refers to Enterprise edition, and **DC** refers to Data Center edition.

For example:

- Windows Server 2008R2 64-bit Standard: WIN0864S
- SQL Server 2008 R2 64-bit Standard: SQL0864S
- Internet Information Services (on Windows Server 2008 R2 64-bit Standard): IIS0864S

NOTE: The name of the Internet Information Services appliance uses the version and edition of Microsoft Windows Server, as opposed to the version and edition of Internet Information Services.

Naming Terminals

Following are some considerations for naming terminals:

- Name terminals based on their role, not their supported protocol
- Use lower case names for terminals (in, out, dbase)

Limit: 3-4 characters.

Naming Boundary Properties

Boundary property names should reflect their purpose. In defining naming conventions for the boundary properties, consider the following:

- If an appliance property is intended for consumption by a specific device, specify the consuming device name in upper case followed by an underscore.
For example: MYDEVICE_service_name
- Use lowercase letters
- When you are creating a new appliance based on an existing appliance and are basing a new property on an existing property name preface that name with the standard abbreviation for the wrapped application. For example:

MYDEVICE_service_name → service name for MYDEVICE product

ABC_service_name → service name for ABC product

MYDEVICE_IN_ip_addr → inbound IP address for the MYDEVICE product instance

ABC_IN_ip_addr → inbound IP address for the ABC product Instance

- When creating an application level property that will be shared by multiple appliances, use the same name you used for the appliance level property. If, however, two or more appliances will be using the same property name but require separate values, you should create two separate application properties with the same name, but preface each one with an abbreviation for the appliance that will consume it. For example:

APPA_service_name

APPB_service_name

If two or more appliances have the same property name, and they both accept the same value, you can create an application property with the same name as the appliance property.

- Use the underscore character to separate words – rather than spaces or dashes.
- To avoid confusion do not use parameter names that are the same or too similar to parameter names used by the product itself

Naming Catalogs and Categories

Appliances and assembly classes are saved to catalogs from which they can be dragged and dropped onto the design canvas to build an application. In addition to the global catalog, each application has its own local catalog. The global system catalog contains all appliance classes that are common for CA AppLogic and are accessible to all applications. The local catalog contains appliance classes that are specific to the application you are editing. If you make a change to an appliance in the global catalog it affects all applications whereas, if you make a change to an appliance in the local catalog the change only affects that application.

Within each catalog, appliance classes are further grouped by category. For example, “database appliances” and “gateway appliances.” When you are wrapping an existing product, consider using the following naming convention for catalog categories:

`<product_name>_<product_release>`

This allows you to differentiate between different releases and different products in an integrated solution. Use the underscore character to link “words” – not spaces or dots.

For example the “MYPRODUCT_2_1” category would contain all appliance and assembly classes for the r2.1 release of MYPRODUCT.

Naming Supporting Documents and Scripts

It is also good practice to establish a set of naming conventions for all related scripts and supporting documents to make it easy to see, at a glance, what product, component and release they pertain to. For example “MYPRODDDB_2_1_class_document” clearly represents the class document for the MYPRODB_2_1 appliance class.

Boundary Property Tips

Following are some additional guidelines for defining boundary properties:

- Order properties according to their importance and likelihood of being changed.
 - List the key functional properties first.
 - Put all mandatory properties near the top.
 - Group related properties together.
 - Keep advanced properties at the end.
- **Tip!** Follow the same order when listing the properties on the class document.
- Try to minimize the total number of properties.
- Do not include any properties that do not make sense in the context of the appliance.
- Avoid properties that are too-advanced and that step outside of the function of the appliance.
- If you can configure multiple modes of the appliance through properties settings consider creating multiple classes of the appliance – each having a dedicated function and properties settings – rather than a single complex appliance.

Terminal Tips

Following are additional guidelines for defining terminal connections:

- Always specify terminal protocols - use ‘any’ only for truly protocol-agnostic terminals.
- Specify terminals that support multiple protocols.

- Order terminals according to their function.
 - Inputs on the left, outputs on the right; feedback terminals, if any, on the reverse side
 - Main data flow terminals are first; auxiliary terminals are last on the side
 - Order the terminals in the data sheet the same way as in the design on the canvas

Volume Tips

Choose volumes based on their role in the appliance from user's perspective. Consider the following:

- Use standard volume names where possible. For example:
 - **Boot:** Instantiable class boot volume
 - **Content:** Placeholder volume, usually reserved for read-only content
 - **App:** Application specific content
 - **Data:** Placeholder volume, usually reserved for read-write data
- Order volumes according to their role, class volumes first.
- Keep volumes small and minimize the number of volumes per appliance.
- Avoid swap volumes for performance reasons.

Script Tips and Directory Structure

If you are creating installation or configuration scripts that need to complete before the application is available to consumers you should modify the `/etc/sysconfig/applogic_appliance` script to kick off those scripts.

Following is an example of the directory structure that was followed for a recent appliance wrapping project.

Directory	Contents
/appliance/init	Parent directory for injected initialization scripts
/appliance/init/<product>	Subdirectory for application specific initialization or configuration scripts. For example, initialization scripts for MYPRODUCT may be kept under the “/appliance/init/myproduct” subdirectory.
/appliance/init/logs	Subdirectory used for initialization log files. Provides a single directory to zip or TAR for support purposes.
/appliance/init/mspk	Subdirectory used for Microsoft product key initialization scripts
/appliance/init/resources	Subdirectory reserved for use by resource files to support location

To minimize impact of possible future changes to the location of the “init” directory, use relative path references when developing scripts and avoid hard-coded absolute paths to content within the “init” structure.

To ensure that processes initiated at startup from “/etc/sysconfig/appllogic_appliance” have sufficient authorization to create or write to files in the /appliance/init” directory structure, permissions must be consistently applied across all components. The following configuration is suggested as the “least open” option that will still ensure proper access to the various initialization processes:

- `chown -R root /appliance/init`
- `chgrp -R SYSTEM /appliance/init`
- `chmod -R 775 /appliance/init`

Standardize Default Passwords

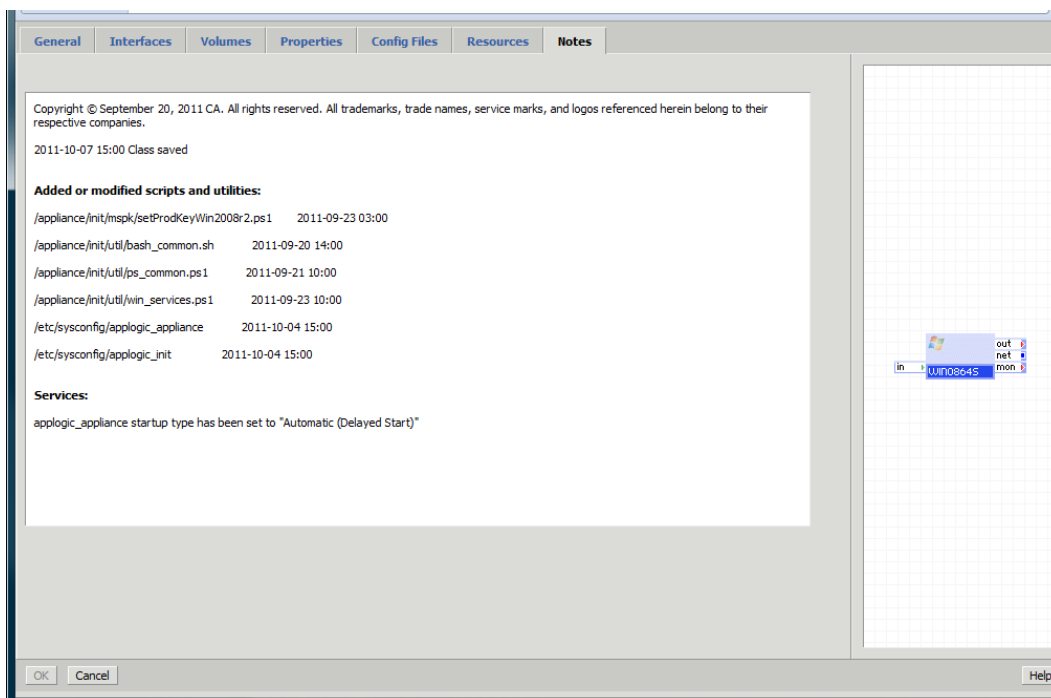
To avoid introducing unnecessary complexity to the project – especially when there are multiple developers or multiple interrelated components involved - establish a default password at the start of the project that will be used for all required user credentials. Consider creating a password that uses both upper and lower case characters, alpha and numeric characters as well as special characters (such as the “@” symbol). Be sure to remind end users to change the default passwords – and provide any details regarding how to do that and where.

Use Annotations on your Design Canvas

When you create a new application, application template or assembly use the CA AppLogic annotation feature to clearly (but briefly) identify the different components and any key details (e.g., port numbers). This makes it easier to demonstrate the purpose served by the individual components – and to highlight, at a glance, any key configuration details. Add Annotations is a right click menu option on the design canvas.

Add Notes when Updating Boundaries

When you update an appliance or singleton embed the appropriate “notes” in the boundary property of the classes and application template to identify what changes have been made. For example:



Chapter 3: Creating a Windows 2008 R2 Appliance

This chapter applies the basic concepts introduced earlier in this document to create a Microsoft Windows 2008 R2 appliance as well as a Windows 2008 Filer appliance which we will require for later examples.

Before you Begin

Following is a list of key documentation topics that are either referenced or relevant to this chapter, along with their quick links:

- “Iso2class: Appliance Distro Creation Utility” section in the *CA 3Tera AppLogic r3.0 Command Line Shell Reference Guide*
http://doc.3tera.com/AppLogic30/en/Cli_Ref/RefIso2Class.html
- “Windows Appliance Installation” section in the *CA 3Tera AppLogic r3.0 Appliance Developer Guide*
http://doc.3tera.com/AppLogic30/en/Developer_Guide/WindowsApplianceInstallation.html
- “Wincfg: Windows Configuration Utility” section in the *CA 3Tera AppLogic r3.0 Command Line Shell Reference Guide*
http://doc.3tera.com/AppLogic30/en/Cli_Ref/RefWinCfg.html

Preparing the Grid

Since the appliances and applications used in the examples in this guide are all Microsoft Windows based there are two preparatory steps that we need to take first.

- [Prepare the grid hardware to run on Windows](#)
- [Import the Filer and System_MS catalog](#)

Note: Both of these steps must be performed by the Grid Maintainer or by another user with sufficient authorization to manage the grid.

Preparing the Grid Hardware to run Windows

In order to run Windows appliances on a CA AppLogic grid, the underlying hardware of each grid node must support Virtualization Technology (VT) and it must be enabled in the BIOS. Since the steps to enable VT vary by manufacturer and version you should consult the individual BIOS manufacturer for the specific instructions for your environment. VT compatibility matrices for both Intel and AMD chips can be found at the following links:

- Intel: <http://ark.intel.com/VTList.aspx>
- AMD: <http://sites.amd.com/us/business/it-solutions/virtualization/Pages/virtualization.aspx>

Import the Filer and system_ms Catalog

Next, copy the following files from the BFC server, and import them into your grid:

- **Sys_Filer_Windows08-<version>.tar**

This file contains the Sys_Filer_Windows08 application which provides filesystem level operations over ntfs08 volumes and is required for the creation of additional ntfs08 volumes. This application is used in the [Creating a Windows 2008 Filer Appliance](#) step. For the purposes of our example, we are going to use version 3.0.1-1.

- **system_ms-<version>.tar**

This file contains the system_ms catalog which includes the “Filer_Windows08” appliance that provides filesystem level operations over ntfs08 volumes and is required to support the creation of additional ntfs08 volumes. This appliance is used in the [Creating a Windows 2008 Filer Appliance](#) section and, for the purposes of our example, we are using version 1.1.3.

The system_ms catalog will be used to store the appliances that are built in this chapter.

These steps only need to be performed once for each grid. If you have already performed them, skip this section and move on to the next section. If not, work with the Grid Maintainer to do following:

1. Import the Sys_Filer_Windows application.

To do this:

- a. SSH into the BFC server(using PuTTY for example), and navigate to the CA 3Tera AppLogic-version Download Directory.

For example

```
/opt/bfc/applogic_versions/<release number>
```

Note: In this example, the release number was 3.0.30.

- b. Copy the file Sys_Filer_Windows08-<version>.tar to the grid controller using following command

```
scp Sys_Filer_Windows08-3.0.1-1.tar <controller IP>:/vol/_impex/
```

- c. SSH to the grid controller(using PuTTY for example) and navigate to /vol/_impex/

- d. un-tar the file using the following command:

```
tar -xf Sys_Filer_Windows08-<version>.tar
```

- e. Import the application by executing the following command:

```
3t app import Sys_Filer_Windows08 Sys_Filer_Windows08
```

The Sys_Filer_Windows08 application should now appear in the list of applications on the CA AppLogic user interface.

2. Import the system_ms catalog.

To do this:

- a. SSH into the BFC server(using PuTTY for example), and navigate to the CA 3Tera AppLogic-version Download Directory. For example

```
/opt/bfc/applogic_versions/<release number>
```

Note: In this example, the <release number> was 3.0.30.

- b. Copy the file system_ms-<version>.tar to the grid controller using following command:

```
scp system_ms-1.1.3.tar <controller IP>:/vol/_impex/
```

- c. SSH to the grid controller (using PuTTY for example) and navigate to /vol/_impex/

- d. un-tar the file using the following command

```
tar -xf system_ms-<version>.tar
```

- e. Import the catalog /system_ms by executing the following command:

```
3t cat import /system_ms system_ms
```

The system_ms catalog imported in the last step contains the IIS03yx4/IIS03yx8 assemblies. These assemblies are designed to include IIS appliances which cannot be distributed due to license requirements. As a result, an error message will be generated every time the editor is opened. To eliminate the error message, either create the missing appliances according to the instructions provided in the [IIS08W/IIS08S/IIS08E/IIS08DC: Installation Reference](#) section of the *CA AppLogic Appliance Developer Guide*, or delete the unused IIS03yx4/IIS03yx8 appliances from the system_ms catalog.

If you choose to delete the appliances *at a later time* you will need to first modify the access to the /system_ms catalog before the classes can be deleted. To do this the **grid maintainer** must do the following:

1. Log into the CA AppLogic Graphical User Interface (GUI) and click the **Applications** tab.
2. Click the **Grid shell** button in the toolbar of the GUI.
3. Type the following command in the Grid Shell:

```
cat put_acl /system_ms local:user:<owner's_user_id>=owner local:group:admin=full
```

4. Execute the following command within the Grid shell for each and every class that needs to be deleted from the /system_ms catalog:

```
class destroy /system_ms:<class_name>
```

We are now ready to build our Windows 2008 r2 appliance.

Creating a Windows 2008 Appliance

In this next section we will walk through the steps to create a new Microsoft Windows Server 2008 R2 64-bit Standard Edition appliance that you can then use as a building block for creating other Windows appliances. The only time you may need to perform these steps again is if you need to create a new appliance for a different edition and/or platform of Microsoft Windows Server 2008 R2.

Note: Although the steps are specific to the 64 bit Standard Edition of Microsoft Windows Server 2008 R2 SP1 (English), they will generally (but not always) apply to other editions and platforms of Microsoft Windows Server 2008 R2.

Step 1: Create the WIN08 Singleton

The first step in creating a new Windows OS appliance is to generate a singleton class for that OS by executing the AppLogic iso2class utility. This utility generates an application from a template that contains a singleton class called iso2class. This singleton will contain two volumes:

- a **boot volume** which includes the ISO image and a
- a second volume which is the target for the OS install

For more information on the iso2class utility consult the following link:

http://doc.3tera.com/AppLogic30/en/Cli_Ref/RefIso2Class.html

To create the Windows appliance singleton do the following:

1. Upload the desired Windows ISO image to your AppLogic grid controller's /vol/_impex volume.
2. SSH into the Grid controller (using PuTTY for example).
3. Begin the installation by executing the following command in a SSH shell:

```
3t util iso2class app_name=Your_App_Name install_size=boot_vol_size  
console_type=graphic iso_volume1=Name_of_windows_iso_image_file virt_options=acpi=1  
cpu=1 mem=2G bw=100M
```

For Example:

```
3t util iso2class app_name=win08_install install_size=16G console_type=graphic  
iso_volume1=en_windows_server_2008_r2_with_sp1_x64_dvd_617601.iso virt_options=acpi=1  
cpu=1 mem=2G bw=100M
```

This creates and starts the win08_install application, booting the singleton iso2class from the specified ISO image.

Notes:

- For the purposes of this example we used a volume size of 16 GB which appeared to provide enough space for the OS installation and subsequent Windows updates, and which resulted in an appliance with approximately 3-3.5 GB of free space at the end of the process. At the time of writing, Microsoft recommends minimum 32 GB of disk space for the installation.
 - In additional testing using a standard ISO that did not include SP1 - 1 GB of memory was allocated to create the appliance and to run Windows Update. The results exhibited some slowness during download and install of the updates that also included SP1. Consequently, if you experience performance issues using these values we recommend increasing the memory and disk space as appropriate.
 - In CA AppLogic, the boot volume of a Windows Server 2008 appliance is of type ntfs08, and ntfs08 volumes may be resized upwards only (in other words, enlarged but not shrunk).
4. In the SSH session, when prompted to specify an Operating System, specify the Operating System you are installing, and press Enter. For example, we specified 8 for Microsoft Windows Server 2008 R2 (64-bit).

```

root@Grid4:~
software and configuration data that are included in the System Catalog.

Do you accept the above terms and condition? (yes/no) yes
APPROLOGIC RESTRICTED AREA
[root@Grid4 ~]# 3t util iso2class app_name=win08_install install_size=16G console_type=graphic iso_volume1=en_windows_server_2008_r2_with_sp1_x64_dvd_617601.iso virt_options=acpi=1 cpu=1 mem=2G bw=100M
Supported OS-es:
[1] Microsoft Windows Server 2003, Datacenter Edition (64-bit)
[2] Microsoft Windows Server 2003, Enterprise Edition (64-bit)
[3] Microsoft Windows Server 2003, Standard Edition (64-bit)
[4] Microsoft Windows Server 2003, Datacenter Edition (32-bit)
[5] Microsoft Windows Server 2003, Enterprise Edition (32-bit)
[6] Microsoft Windows Server 2003, Standard Edition (32-bit)
[7] Microsoft Windows Server 2003, Web Edition
[8] Microsoft Windows Server 2008 R2 (64-bit)
[9] Microsoft Windows Server 2008 (32-bit)
[10] Linux (32-bit)
[11] Linux (64-bit)
[12] Sun Solaris 10 (32-bit)
[13] Sun Solaris 10 (64-bit)
[14] Other (32-bit)
[15] Other (64-bit)
Please specify an OS number and press [enter]: █

```

Although the SSH session will prompt you to press Enter when you have finished installing, DO NOT press Enter, or close this window at this time. Leave the window open as you will come back to this later on. For now, proceed to the next step..

```

root@Grid4:~
Starting application win08_install
main.iso2class started

Application win08_install started successfully

*****
*****

Please access the console of the singleton iso2class by selecting it within
the application editor and then using the Appliance drop-down menu to open
the graphic console.

If you are installing a Linux OS, do NOT install an APK, this will be done
by hvm2pv. See the hvm2pv documentation for details.

If you are running the install on an ESX-based grid or intend to use an
unmanaged appliance (no APK is installed) on an ESX-based grid, it is
recommended to install vmware-tools in the appliance. vmware-tools is
needed by unmanaged appliances so that they can shutdown in a timely manner.
vmware-tools can also be used in managed appliances to provide better support
for accessing graphical desktops.

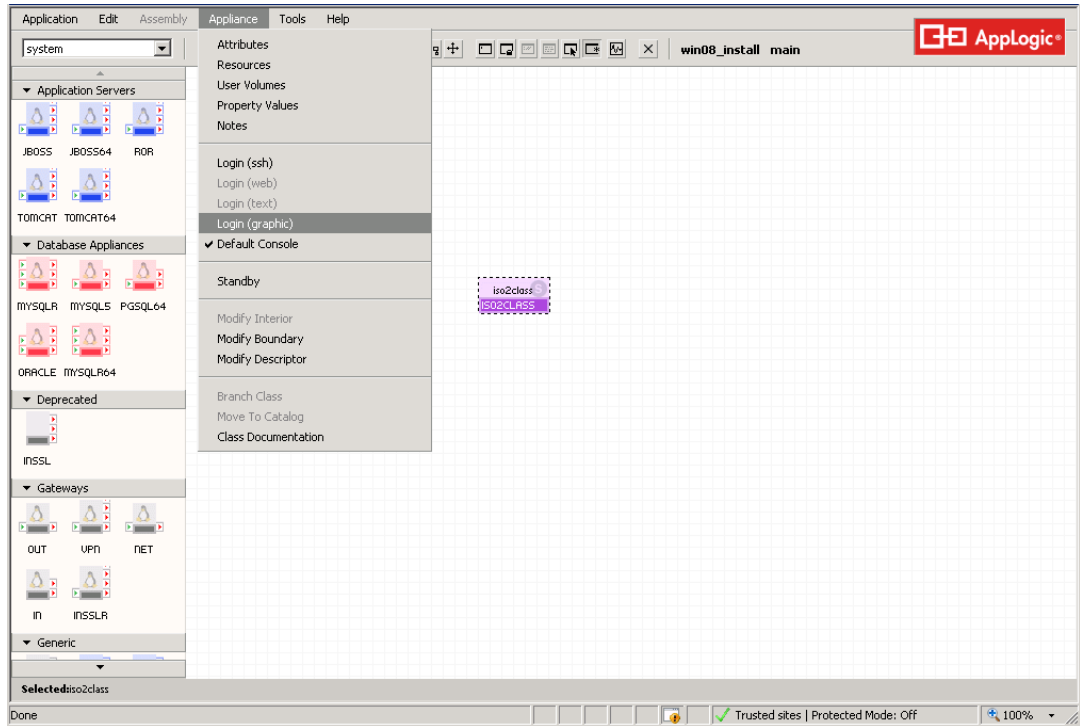
When you have finished installing from iso_volume1 and are ready to reboot
so that remaining volume(s) become available, press [Enter]

```

5. Use one of the following methods to access the graphic console of the singleton:
 - Select the application within the application list in the CA AppLogic GUI and click the icon for **Login (graphic)** within the toolbar.

Dashboard	Applications	Logs	Support
<div style="border: 1px solid #ccc; padding: 5px;"> <div style="display: flex; justify-content: space-between; border-bottom: 1px solid #ccc;"> Oblicore_Win08linux_2_MSP (template) Stopped CA Oblicore 7.0SP1 on Applogic v3.0 - (v. 2.5.0.1) 2.20 6.50G 1.00G </div> <div style="display: flex; justify-content: space-between; border-bottom: 1px solid #ccc;"> SugarCRM_r14 (template) Stopped Fully featured, scalable CRM Application, based on SugarCRM's Sugar Open Source 5.2.0 (v5.2.0-10) 2.15 2.94G 1.55G </div> <div style="display: flex; justify-content: space-between; border-bottom: 1px solid #ccc;"> Sys_Filer_Linux (template) Stopped Linux Filer Application (v4.0.2-1) 0.05 512.00M 1.00M </div> <div style="display: flex; justify-content: space-between; border-bottom: 1px solid #ccc;"> Sys_Filer_Solaris (template) Stopped Solaris Filer Application (v3.0.4-1) 0.05 512.00M 1.00M </div> <div style="display: flex; justify-content: space-between; border-bottom: 1px solid #ccc;"> Sys_Filer_Windows08 (template) Stopped Windows08 Filer Application (v3.0.1-1) 0.25 512.00M 1.00M </div> <div style="display: flex; justify-content: space-between;"> win08_install Running (less than an hour) Generated by the iso2class utility. 1.00 1.00G 1.00M </div> </div>			

- Open the application in the CA 3Tera AppLogic editor and select the singleton iso2class. Use the pull-down Appliance menu to select **Login (graphic)**.



Note: If you get a message indicating “Failed loading application ‘win08_install’. Permission denied: cannot load application” you will need to work with the grid administrator to modify your permissions. See [What Happens If I get a Permission Denied Message](#) for more details.

6. Respond to the prompt(s) regarding source and authenticity.

Note: Your response will depend on your particular security policies and settings. For example, in our environment when prompted to continue because the web site’s certificate could not be verified, we clicked **Yes**.

Once the graphical console appears, the Windows 2008 Server installation interface should be visible and we can proceed with the standard OS install keeping the following in mind:

- For the purposes of this example, we have selected Windows Server 2008 R2 Standard (Full Installation) as the Operating System *Type*.
- When asked where to install Windows, accept the default option and click **Next**.
- The singleton will reboot during installation, rendering the graphical console inoperative. When this happens, simply close and re-open the graphical console.
- On a rare occasion, when using Mozilla's Firefox to access the graphical console, you may receive an error indicating that the Application is not started or the graphical console is misconfigured. If this should happen simply close and re-open Firefox.
- For the purposes of this example we are assuming a full Installation of Windows with its more complete graphical interface. If you are installing a Server Core installation you will need to adjust your keyboard and mouse actions to accomplish these same steps.

- During Windows installation on a VMware grid, due to absence of VMware tools, the mouse may not function as expected. The following keyboard shortcuts are useful:
 - To navigate between buttons use Tab key and / or arrow keys
 - To select use Space bar or Enter key
 - To access the file menu of the selected window use Alt+f key

The singleton will reboot itself when the installation is complete. When this happens:

1. Close and re-open the graphical console.
You will be prompted to set a new password.
2. Type in your new password and press **Enter**.
Windows will set a new password for the Windows Administrator user and display a message indicating that the Administrator password is set successfully.
3. Click **OK** to the message
Windows will login to the Windows Administrator desktop.

What Happens If I Receive a Permission Denied Message?

If you get a “Failed loading application ‘win08_install’. Permission denied: cannot load application” message when you open the application you will need to work with your **grid administrator** to modify the application permissions, assigning you as the owner of the application and giving the AppLogic group that you belong to, “full” permissions to the newly created application. The grid administrator does the following:

1. Right-click on the application within the application list in the CA AppLogic GUI and click **Configure**.
2. Click the **Security** tab within the Application Configuration window.
3. Specify your user name for the owner of the application.
4. Add your username or the group name (for which you are a member of) to the permissions list with full access and local scope by clicking the **Add** button and providing an appropriate value for each field.

For example:

- **Name:** admin
- **Access:** full
- **Type:** group

Note: If specifying a group name, specify the type to be group; otherwise, specify user.

- **Scope:** local

5. Click **OK**
You should be able to open the application editor now.

Step 2: Configure and Update Windows

Once the OS has been installed on the singleton the next step is to modify the default configuration and execute Windows Update to ensure that the latest patches and security fixes - which may not be part of the ISO used for the appliance creation - are included.

For ESX Grids only:

If you are creating Windows appliances on an **ESX grid**, you need to first install the VMware tools by doing the following:

Note: If you are not creating Windows appliance on ESX grid skip this step:

1. Stop the application
2. In a separate SSH shell, execute the following:

```
3t vol copy _GLOBAL_RO:vmware_tools_windows <APPNAME>:vmware_tools_windows
```

Where **<APPNAME>** refers to the application name. In this example **win08_install** was used.
3. Open the application editor in a browser by clicking on the application within the CA AppLogic GUI.
4. Right-click on the singleton appliance 'iso2class' and click **User Volumes**
5. Click in the field for under the column **App Volume**, for the volume **iso_volume1** and select the volume **vmware_tools_windows**.
6. Click the **Save** button in the Application Editor, and close the Application Editor.
7. Right-click the application and click **Start** in the context menu to start the application.
8. With the application selected, login through the graphical console by clicking the **Login (graphic)** button in the toolbar of the CA AppLogic GUI.
9. Press the **Ctrl+Alt+Del** button on center top of the graphical console window and login to the desktop as the Windows Administrator user.
10. Click inside the graphical console window anywhere
11. Using 'tab' and 'arrow keys' navigate to "start task manager"
12. Using 'tab' navigate to "new task" and hit 'spacebar'
13. Type the following command to execute VMware tools installation silently:

```
D:\setup.exe /S /v /qn
```

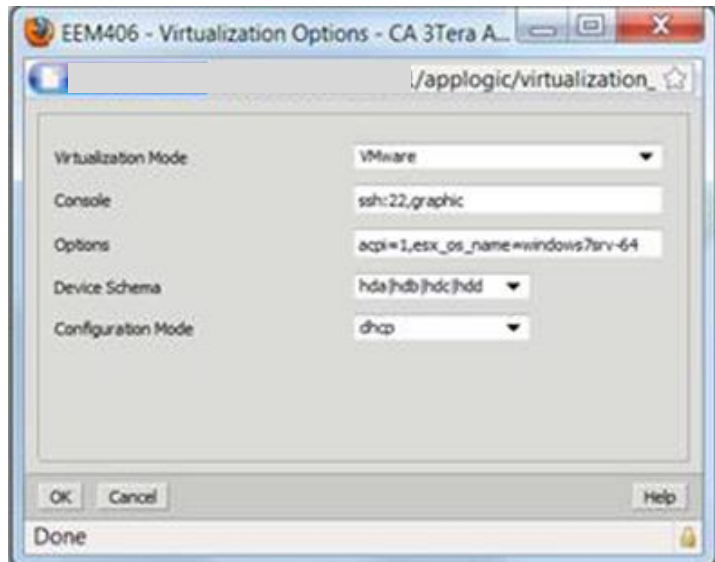
The installation will reboot the appliance automatically. Wait patiently.
14. After reboot reopen the graphical console by clicking the **Login (graphic)** button in the toolbar of the CA AppLogic GUI and login to the appliance.

For appliances that will be migrated to an ESX grid:

If you create an appliance on a non-ESX grid but plan to port it to an ESX grid in the future you will need to make the following configuration change to enable installation of VMware tools on the appliance once it is ported:

1. Stop the application.
2. Open the Application Editor in a browser by clicking on the application within the CA AppLogic GUI.
3. Right-click on the singleton appliance 'iso2class' and click **Modify Boundary**
4. On the General tab, find the **Options** button in the Advanced section
5. In the pop-up window, select *VMware Virtualization Mode*.
6. Under the Options section add the 'esx_os_name' setting with its corresponding value.

For example, for a 64-bit Windows 2008 R2 appliance, the value should be windows7srv-64. For a full list of values, click [here](#).



7. Click **OK** to close the window, and click **OK** again on the Modify Boundary window.
8. Start the application and reopen the graphical console by clicking the **Login (graphic)** button in the toolbar of the CA AppLogic GUI and login to the appliance.

You are now ready to make the configuration changes.

Configuration Changes

The following changes apply to both ESX and non-ESX grids.

Set the **screen resolution** and **screen saver timeout** by doing the following:

1. Right-click on an empty area of the desktop and select **Screen Resolution**.

2. From this interface set the screen resolution to 1024x768 and click apply. When prompted, choose the option to keep your changes.
3. In the Windows toolbar, click **Display**, and then click **Change screen saver**.
4. Set the Screen saver to **(None)**, and click **OK**.

Next, **disable the page file** by doing the following:

1. Open the Windows Control Panel and click **Appearance**.
2. Under **Folder Options**, click **Show hidden files and folders**.
3. In the **Folder Options** dialog, select **Show hidden files, folders, and drives** and uncheck **Hide protected operating system files (Recommended)**.
4. When prompted to confirm to display these files, click **Yes**.
5. Click **OK** in the **Folder Options** dialog.
6. Click on the **Control Panel Home** link, and click **System and Security**.
7. Click the **System** link.
8. Click the **Advanced system settings** link.
9. Click on the **Settings...** button for **Performance**.
10. Click the **Advanced** tab in the **Performance Options** dialog, and click the **Change...** button under **Virtual memory**.
11. Clear the check box marked **Automatically manage paging file size for all drives** and select **No paging file**.
12. Click the **Set** button. When prompted **...Do you want to continue**, click **Yes**.
13. Click the **OK** button in the **Virtual Memory** dialog.
14. Click the **OK** button in the **Performance Options** dialog.
15. Click the **OK** button in the **System Properties** dialog.

Next, **disable power management hibernation** support by opening a Command Prompt and executing the following command:

```
powercfg -h off
```

Disable IE enhanced security by doing the following:

1. Open Server Manager by clicking the **Server Manager** button in the taskbar at the bottom of the desktop.
2. With **Server Manager (...)** selected, click the **Configure IE ESC** link in the **Security Information** section.
3. Turn IE enhanced security off for both **Administrators** and **Users**.
4. Click **OK** when done.

Change the **computer name** by doing the following:

1. With **Server Manager** still open, and with **Server Manager (...)** selected, click the **Change System Properties** link in the **Computer Information** section.
2. Click the **Change...** button to rename the computer.
3. Change the Computer Name and verify that the **Workgroup** option is selected, and click OK. In this example the computer name was set to **WIN0864S**.
4. Click **OK** when prompted that the computer must be restarted.
5. Click the **Close** button in the **System Properties** dialog.
6. Click **Restart Now**.

After a minute re-open the graphical console by clicking the **Login (graphic)** button on the toolbar of the CA AppLogic GUI with the application selected, and login to the Windows desktop.

Note: To send CTRL+ALT+DELETE, move the mouse towards the top of the screen and click the **Ctrl+Alt+Delete** button.

Next, clean up the leftover pagefile if it is present. To do this:

1. Delete the file C:\pagefile.sys if it is present.
2. Open the Windows Control Panel and click **Appearance**.
3. Under **Folder Options**, click **Show hidden files and folders**.
4. In the **Folder Options** dialog, select **Don't show hidden files, folders, or drives**, check **Hide protected operating system files (Recommended)**, and click **OK**.

Note that the singleton has two **network interfaces**:

- Local Area Connection – the external interface
- Local Area Connection 2 - the internal interface.

Open a Command Prompt and ping www.ca.com.

If you receive a response then install the [high priority and optional Windows updates](#).

If you do not receive a response, you first need to configure the external interface of the singleton to provide access to the internet by doing the following:

1. Open **Server Manager** by clicking the **Server Manager** button in the taskbar at the bottom of the desktop.
2. With **Server Manager (...)** selected, Click **View Network Connections** in the **Computer Information** section.
3. Double-click **Local Area Connection**
4. Click the **Properties** button.
5. Select **Internet Protocol Version 4 (TCP/IPv4)** and click the **Properties** button.

6. Select **Use the following IP address** and enter usable values for your grid (IP Address, netmask, gateway, DNS server). Click **OK** when done.
7. Click the **Close** button in the **Local Area Connection Properties** dialog.
8. Click the **Close** button in the **Local Area Connection Status** dialog.

Next, install high priority and optional **Windows updates** by performing the following steps.

Note: If the installation gets stuck or appears to run very slowly, you may need to restart the appliance creation process and increase the memory or disk space (see [earlier note](#)). You may also need to restart the process if the ssh session running the iso2class utility gets stuck and you are unable to get to the step where we install the APK and make the appliance a managed appliance..

■ For **32-bit versions of Windows**, perform the following steps:

1. Open Internet Explorer and click the **Tools** menu.

Note: If prompted to setup Windows Internet Explorer, click the **Ask me later** button.

2. Click **Windows Update**
3. Select the option **Turn on Now and install updates**.
4. Reboot when prompted. Close the graphical console, and, after a minute, re-open the graphical console.
5. Repeat the above procedures until there are no more updates to install.

■ On **64-bit versions of Windows**, perform the following steps:

1. Open Windows Control Panel.
2. Click the **System and Security** link.
3. Click the **Windows Update** link.
4. Click the **Change settings** link.
5. Select **Download updates but let me choose to install them** from the **Important updates** drop-down list, then click **OK**.

Windows will automatically check for updates.

6. Reboot when prompted.
7. Close the graphical console, and, after a minute, re-open the graphical console.
8. Repeat the above procedures until there are no more updates to install.

Step 3: Activate Windows

The next step is to activate Windows, however, depending on how you plan to implement licensing and distribution for the appliance it may make sense to defer activation to a later time. For a discussion of how licensing decisions can impact Windows activation refer to the [Licensing: Microsoft Windows Server](#) section. For details on how to activate Windows at a later time, see [Activate Windows Later](#).

To activate Windows on a **32-bit system** :

1. Open the Windows Control Panel
2. Click on the **System** link.
3. Click the **Change product key** link and complete the **Windows Activation** wizard.

To activate Windows on a **64-bit system**:

1. Open the Windows Control Panel
2. Click the **System and Security** link.
3. Click the **System** link.
4. Click the **Change product key** link and complete the Windows Activation Wizard.

Step 4: Post Install Configuration

The following post install steps for configuration of a Windows singleton are included in the CA AppLogic product documentation and are highly recommended to ensure maximum re-usability and stability of the appliance.

First, **disable external network access** by doing the following:

1. Open Server Manager by clicking the **Server Manager** button on the taskbar at the bottom of the desktop.
2. With **Server Manager (... selected**, click on the **View Network Connections link** in the **Computer Information** section.
3. Double-click **Local Area Connection**.
4. Click the **Properties** button.
5. Select **Internet Protocol Version 4 (TCP/IPv4)** and then click the **Properties** button.
6. Select **Obtain an IP address automatically** and **Obtain DNS server address automatically**, and click the **OK** button.
7. Click the **Close** button in the **Local Area Connection Properties** dialog.
8. Click the **Close** button in the **Local Area Connection Status** dialog.
9. Close the **Network Connections** window.

Next, with Server Manager still opened, **disable the Windows Firewall** by doing the following:

1. With **Server Manager** (... selected, click the **Go to Windows Firewall** link in the **Security Information** section.
2. Click on **Properties** under the **Actions** column.
3. Set the firewall state to Off for the Domain Profile, the Private Profile and the Public Profile.
4. Click the **OK** button.

With Server Manager still opened, **disable automatic updates** by doing the following:

1. With **Server Manager** (... selected, click the **Configure Updates** link in the **Security Information** section.
2. Click the **Change settings** link.
3. Select **Never check for updates** option in the **Import updates** drop-down list.
4. Click the **OK** button.
5. Close the **Windows Update** window.

Next, **disable password complexity requirements** if set, by doing the following:

1. Navigate to **Start > Administrative Tools > Local Security Policy**
2. Expand the **Account Policies** node.
3. Click the **Password Policy** node.
4. Double-click **Password must meet complexity requirements**.
5. Select **Disabled** on the **Local Security Setting** tab, and click the **OK** button.
6. Close the **Local Security Policy** window.

With Server Manager still opened, set the **Administrator password** to never expire by doing the following:

1. Expand the **Configuration** node.
2. Expand the **Local Users and Groups** node.
3. Click on the **Users** folder.
4. Double-click the **Administrator** user.
5. Check **Password never expires**, and click the **OK** button.

Now, **disable the shutdown event tracker** by doing the following:

1. Open **Start > Run...**
2. Type **gpedit.msc** in the **Run** dialog, and click the **OK** button.
3. Expand **Computer Configuration > Administrative Templates** and click on the **System** folder.
4. Scroll down on the right pane and right-click on **Display Shutdown Event Tracker** and click **Edit**.

5. Select **Disabled** and click **OK**.
6. Close the **Local Group Policy Editor** window.

Next, set the **default Windows disk device timeout value** to 60 seconds if it is not already set to 60 seconds by doing the following:

1. Click **Start > Run...**
2. Type **regedit.exe** in the **Run** dialog, and click **OK** to open the registry editor.
3. Navigate to the key **HKEY_LOCAL_MACHINE/System/CurrentControlSet/services/Disk**.
4. Add or modify the **TimeOut Value** as follows:

If there is already an entry name **TimeOutValue** perform the following steps:

- a. Set the timeout to 60 seconds as required by double-clicking **TimeOutValue**
- b. Select **Decimal**, and changing the **Value data** to 60.
- c. Click **OK** if changes were made; otherwise, click **Cancel**.

Note: If the value is already set to 60, then it is ok to leave **Hexadecimal** selected.

If an entry for **TimeOutValue** does not exist, perform the following steps:

- a. Right-click on **Disk** in the left pane and select **New > DWORD (32-bit) Value**.
- b. Type **TimeOutValue** followed by pressing the enter key. Right-click on **TimeOutValue** and select **Modify....** Set the **Base** to **Decimal** and enter 60 for the value.
- c. Click **OK**.

Step 5: Configure the Singleton to be Fully Managed by CA AppLogic

The next step is to configure the singleton to be “fully managed by CA AppLogic.” This is done by installing the Windows Server MSI, which installs the Windows Appliance Kit (APK) and it enables the following interactions between CA AppLogic and the appliance:

- The ability for the singleton/appliance to communicate with the grid controller by sending events if problems may occur.
- The ability to auto-configure the network interfaces of the singleton/appliance.
- The ability to obtain property values from the boundary of the singleton/appliance.

The Windows Server MSI is located on one of the volumes of the singleton and it is installed by doing the following:

1. Shutdown the OS and continue the iso2class process by performing the following steps:
 - a. Open a Command Prompt, execute the following command:

```
shutdown -s -t 0
```

Note: If this is a 64 bit operating system see the [Notes regarding 64 bit Operating System](#) section for information on copying the shutdown.exe if it is not already present.

- b. Wait until Windows has shut down.
- c. Return to the SSH session from the section [Creating the WIN08 Singleton](#), step 5 within that section, and press enter, indicating that the Windows installation has complete. When prompted if you are certain, type 'Y' and press enter.

Note: This will restart the application.

- d. When prompted to continue after finishing the installation, **DO NOT** press Enter or close this window. Leave this window open and continue on to the next step.

```

root@Grid4:~
Starting application win08_install...
Building application...
  Configuring application...Done
Loading application...
Scheduling application...

Starting application win08_install
  main.iso2class started

Application win08_install started successfully

*****
*****

Please access the console of the singleton iso2class and continue with
the OS installation.

If you are installing a Windows OS, be sure to include security updates,
service packs, resource kit tools, and any other software you need on every
appliance derived from this image. Then install the Windows Server msi.
See the iso2class documentation for details.

When you have finished with the installation, press [Enter] to continue.
  
```

- 2. After a minute, re-open the graphical console by selecting the application, and clicking the **Login (graphic)** button in the toolbar of the CA AppLogic GUI
- 3. Login to the Windows desktop using the administrator credentials.

Note: If the graphical console starts up in the System Recovery Options dialog, do the following

- a. Select Next
- b. login as Administrator
- c. Restart
- d. After a minute repeat the previous step (step 2)

- 4. Install the Windows Server msi:

Note: The Windows Server msi is available on the E drive of the singleton.

- a. Open a Command Prompt
- b. Execute the following command to navigate to the E drive:

E:

- c. Execute the following command:

```
Server_windows-<version>.msi
```

Note: Version **2.0.2-1** was used in this example.

- 5. Complete the basic installation by doing the following:

- a. Enter the following command in the command prompt:

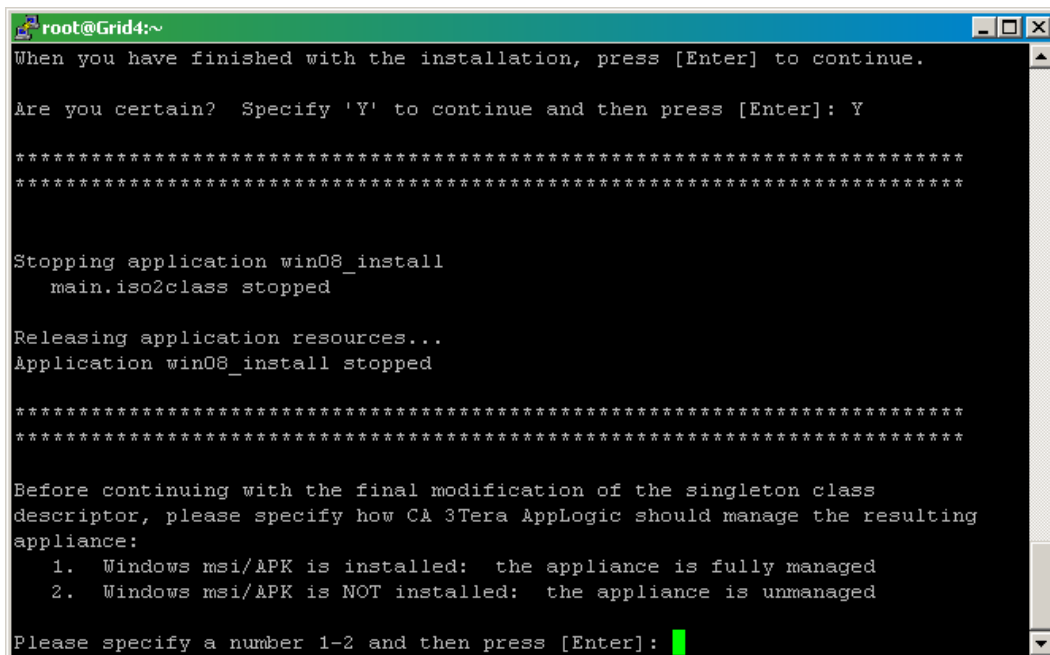
```
shutdown -s -t 0
```

- b. Wait for Windows to shutdown.

- c. Return to the SSH session from step 1d of this section. Within the SSH session running the iso2class utility: Press the **<enter>** key and then type **Y** to confirm installation has finished, and press the **<enter>** key.

- d. When prompted in the SSH session, type **1** to specify that **Windows msi/APK is installed: the appliance is fully managed**, and press the **<enter>** key.

The iso2class stops the application, changes the singleton boundary to that of a generic server appliance, and re-starts the application (the appliance will enter maintenance mode during the restart).



- e. Verify that you see the message “iso2class has completed its operation.”

```
root@Grid4:~
Starting application win08_install...
Building application...
  Configuring application...Done

Loading application...
Scheduling application...

Starting application win08_install
  main.iso2class entered maintenance state
  main.iso2class started

Application win08_install started successfully

*****
*****

iso2class has completed its operation. The resulting singleton iso2class
in application win08_install uses HVM virtualization (hardware emulation).

For more information, see the on-line reference at:
http://doc.3tera.com/AppLogic30/RefIso2Class.html

APPLOGIC RESTRICTED AREA
[root@Grid4 ~]#
```

Since the Windows Administrator password was set to a different password during this process your next step is to reset the password.

Note regarding 64-bit Windows Operating Systems

Depending on which Windows release you are using, if this is a 64-bit operating system you may need to use Windows Explorer to copy **shutdown.exe** from **C:\WINDOWS\system32** to **C:\WINDOWS\SYSWOW64** if it is not already present. If Windows Explorer refuses to copy but permits you to move the executable then perform the following steps:

- 1. Move shutdown.exe to C:\WINDOWS using Windows Explorer.
- 2. Double-click Cygwin on the desktop and execute the following command:
- 3. cp -p /cygdrive/c/WINDOWS/shutdown.exe /cygdrive/c/WINDOWS/system32/shutdown.exe
- 4. NOTE: This command actually copies the file into C:\WINDOWS\SYSWOW64.
- 5. Use Windows Explorer to move C:\WINDOWS\shutdown.exe to C:\WINDOWS\system32\shutdown.exe.

Note: In our testing on Windows Server 2008 R2, this step was not needed as shutdown.exe existed in both places. This step may be needed on other versions/editions of Microsoft Windows Server.

Resetting the Windows Administrator Password

By default, AppLogic scripts set the administrator password to be a random password after the msi file is installed. Therefore, you will need to perform the following action to reset the password:

Note: Information on disabling this functionality is provided later in this document.

1. Click the **Login (ssh)** button either in the CA AppLogic GUI toolbar with an application selected, or in the toolbar of the Application Editor with the appliance/singleton selected.
2. Execute the following command:

```
net user <user_to_reset> <new_password>
```

For example:

```
net user administrator myNewPASS123
```

Step 6: Optional Configurations

Following are some additional configuration changes you may want to make – depending on your specific appliance requirements. They are optional. To make any of these changes you first need to login to Windows by doing the following:

1. Select the application in the CA AppLogic GUI.
2. Click the **Login (graphic)** button in the toolbar.
3. Login to Windows with the Windows Administrator’s credentials.

Note: If having trouble logging in, see the section [Resetting the Windows Administrator Password](#)

Activate Windows Later

Depending on how you choose to manage licensing for the Windows appliance it may make sense to delay activation of Windows until a later time (see the [licensing discussion](#) earlier in this guide for an example). If so, you will need to do the following:

- Disable Auto-Activation so that Windows will not try to activate itself.
- Deactivate Windows if Windows has already activated itself
- Create a script to automatically activate Windows, and execute upon startup.

Details on performing these steps follows. Note that these steps require changes to the Windows registry.

To **disable auto-activation** for Microsoft Windows Server 2008 r2 do the following:

Note: It is highly recommended to backup the registry before making any changes it could lead to unintended results.

1. Click Start > Run... and execute **regedit.exe** to open the registry and navigate to **HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\SoftwareProtectionPlatform\Activation**
2. Double-click the DWORD **Manual** and set the value data to **1** (verify that **Hexidecimal** is selected as the Base).

3. Click **OK**, and close the **Registry Editor** window.
4. Restart Windows.

Note: The graphical console will become unusable, so you will need to login again by selecting the application within the CA AppLogic GUI, and clicking the **Login (graphic)** button in the tool bar to open a new graphical console. Login to Windows as usual.

To **deactivate a Microsoft Windows Server** that has already been activated, perform the following steps:

1. Open a Command Prompt and execute the following command:

```
s1mgr.vbs /upk
```

Note: Click the **OK** button when the message **Uninstalled product key successfully** is displayed.

2. Restart Windows.

Note: The graphical console will become unusable, so you will need to login again by selecting the application within the CA AppLogic GUI, and clicking the **Login (graphic)** button in the tool bar to open a new graphical console. Login to Windows as usual

If you decide to deactivate Windows or disable auto-activation, you should also consider how you want your end users to activate it later. Although this could be done manually, you may want to use scripts to **provide a standard way for end users to both license and activate Windows**. One approach for creating this script is discussed in [Appendix A](#). The Microsoft License Key Injection script utilizes two properties – win2008r_prodkey and kms_server_port - that are defined on the appliance boundary to inject the appropriate licensing details for the appliance. When used, this script is stored in the C:\appliance\init\mspk folder and generates log messages to a file stored in the C:\appliance\init\logs folder. It also utilizes a common utilities script that is stored in the C:\appliance\init\util directory). Further information about this script, as well as the commands that can be called to license Microsoft Windows server, see [Appendix A](#).

If you plan on using scripts to license Windows – either the field developed Microsoft License Key Injection utility or your own custom script – this is the time to create and implement those scripts.

Disabling Auto Host Name Generation

By default, the hostname that is assigned to the appliance will be the same as the instance name of the appliance. If the instance name is changed, the Windows hostname will change as well without any warning. There are several reasons why this feature may be undesirable, including:

- If the appliance resides in an AppLogic Assembly (see the [AppLogic Assemblies](#) section later in this document for more details), then the hostname of the appliance will also include the instance name of the assembly, which means a longer hostname.

- Negative or unexpected impact on other software installed on the appliance. Developers will need to consider the possible effects of a hostname change before renaming the instance.

To disable this feature and enable end users of the appliance to set the hostname using the traditional method, do the following:

Note: Even if the steps are not performed now, they can still be performed at a later date.

1. Start the appliance, if it has not already been started by right clicking the application in the CA AppLogic GUI and selecting **Start** in the context menu.
2. Log into the appliance using the graphic console by:
3. Click the application to open the Application Editor
4. Select the singleton/appliance to log into
5. Click the **Login (graphic)** button in the toolbar of the Application Editor.
6. Provide the appropriate Windows credentials.
7. Double click the **Cygwin** application on the Desktop.
8. Using an editor like VI, create a new file(if it doesn't already exist) called
`/etc/sysconfig/applogic_init`
9. Edit the file and add the following line to the end of the file:

```
APK_HOSTNAME_UPDATE=no
```

10. Save your changes.

Tip! For information on using the field developed Windows Host Name Update utility to simplify this process, see the CA AppLogic Enablement Kit, which is available through the AppLogic Partners User Group on Cloud Commons (www.cloudcommons.com)

Disable Random Password Generation

For Windows based appliances, every time an application or component name changes, a random password is generated for the Windows “Administrator” account. When that happens, you will need to SSH into the appliance and [reset the Windows Administrator's password](#). Random password generation can be disabled by setting the **gen_pwd** value to “0” in the **/appliance/appliance.sh** file. To do this:

1. Verify that the application is started, if not, right-click the application in the CA AppLogic GUI and click **Start** in the context menu.
2. Log into the appliance using the graphic console by:
3. Click the application to open the Application Editor
4. Select the singleton/appliance to log into
5. Click the **Login (graphic)** button in the toolbar of the Application Editor.

6. Provide the appropriate Windows credentials.
7. Double-click the **Cygwin** application on the Desktop.
8. In an editor like VI, edit the **/appliance/appliance.sh** file, locate the line where **gen_pwd** is being set, and change the value to 0:

```

#* ----- appliance.sh - Appliance controlling script ----- *
#* Copyright (C) 2005, 2011 CA. All rights reserved. *
#* Use of this notice does not imply publication or disclosure. *
#* This software contains CONFIDENTIAL and PROPRIETARY information *
#* constituting valuable TRADE SECRETS of CA, Inc., which may be disclosed *
#* by CA, Inc., only under strict limitations on its use and *
#* permitted in writing by CA, Inc. *
#* In addition, any hard-copy, printout or other tangible expression of this *
#* software must include on every page thereof the copyright notice contained *
#* herein and the phrase "CA CONFIDENTIAL - HIGHLY PROPRIETARY". *
#* ----- *

source /var/run/applogic/appliance.sh

PWDFILE=/appliance/passwd.stamp

case "$1" in
start)
# determine if Administrator's password needs to be changed
gen_pwd=0
current_name="$__APP_NAME:$__COMP_NAME"
if [ -f $PWDFILE ]; then
old_name=`cat $PWDFILE`

```

9. Save the file.

Note: This tip works only for fully managed Windows appliances

Enable Remote Desktop

At this point, the only way to log into the singleton/appliance is through the graphic console within the CA AppLogic GUI. To allow end users to log into the singleton/appliance without requiring them to log into the CA AppLogic GUI, you can enable Remote Desktop by doing the following:

1. Open the Windows Control Panel.
2. Click the **System and Security** link.
3. Click the **Allow remote access** link under **System**.
4. Specify one of the two options that best fits the security policies of your company in the **System Properties** dialog, and click **OK**.

Note: When notified that a Remote Desktop Firewall exception will be enabled, click the **OK** button.

Step 7: Modify the Boundary and Instance Settings

In this step we will modify the boundary of the singleton to prepare it to become an appliance. To do this:

1. Stop the application by right-clicking the application in the CA AppLogic GUI and clicking **Stop** in the context menu.

2. Open the Application Editor by clicking the application in the CA AppLogic GUI.
3. Right-click on the singleton and click **Attributes**. Change the instance name to WIN0864S, and click the OK button.
4. Right-click on the singleton and click **Modify Boundary**:
5. Make the following changes:

On the **General** tab...

- Change the class name to **WIN0864S**.
- Change the Category to **Operating Systems**.

NOTE: Categories help organize where appliances appear within a CA AppLogic Catalog. All appliances that provide a similar function need to have the same category so they appear together within the catalog. Note that if you try to add an appliance to a catalog that contains another appliance with the same name, there will be a conflict, even if the two appliances belong to two different categories within the same catalog.

- Enter an appropriate description. For example:
“Windows Server Appliance - based on Windows Server 2008 R2 SP1 64-bit Standard Edition”
- Set the documentation URL to point to the following location:
http://doc.3tera.com/AppLogic30/en/Catalog_Ref/CatGenericWindows.html
- Change the Color to **blue**.
- Optionally change the Size.
- Set the OS Icon to Windows.

On the **Interfaces** tab verify that **External Interface** is unchecked

Click the **Properties** tab and review the list of configuration properties.

For the purposes of the examples in this document we are using the configuration scripts documented in [Appendix A](#) to manage to manage licensing for this appliance. These scripts require the addition of two new properties to the appliance. Following are the instructions on how to add these properties. If you are not using these scripts, skip to the next step.

- a. Click the **Add** button.
- b. Click in the field under the **Name** column, and change the name to **win2008r2_prod_key**.

The licensing script will expect the product key for Microsoft Windows Server for the value of this property. If the property value is left blank, then the licensing script will skip trying to license Microsoft Windows Server.

- c. Repeat steps a and b to add the property **kms_server_port**.

The licensing script will expect the Key Management Server and Port number to use to license Microsoft Windows for the value of this property. The format of this value is `<kms_server_name>:<port>`. If this property value is left blank, then it is assumed that a key management server will not be used.

Click the **Resources** tab and modify the resources as follows:

■ **CPU (num)**

- Min: 0.25
- Max: 4
- Default: 1

■ **Memory (bytes)**

- Min: 512M
- Max: 4G
- Default: 2G

■ **Bandwidth (bits/sec)**

- Min: 1M
- Max: 2G
- Default: 100M

d. Click **OK**, and save the application when prompted.

6. Right-click the application within the CA AppLogic GUI, and click **Start** in the context menu.
7. With the application selected in the CA AppLogic GUI, click the **Login (graphic)** button to open the graphical console.
8. Login to Windows by pressing the **Ctrl+Alt+Del** button at the top of the graphical console, and providing the appropriate credentials.

Step 8: Prepare the Singleton to be Added to a Catalog

Adding the singleton to the catalog enables you to create a new appliance that can be re-used multiple times to create additional instances of the appliance— and as the basis for creating additional appliances. Before adding the singleton to a catalog, however, there are some cleanup steps you should take to prepare it so that future users do not see a record of any changes that you may have made during the creation process. These changes include:

- [Modifying the TCP/IP settings](#)
- [Verifying Optional Configuration settings](#)
- [Resetting the administrator password](#)
- [Clearing remote desktop history cache](#)
- [Clearing browser history](#)

- [Collapsing all nodes in registry](#)
- [Collapsing all nodes in the Server Manager display](#)
- [Deleting unnecessary files and logs](#)
- [Clearing event viewer entries](#)
- [Clear Start menu cache](#)
- [Emptying the Recycle bin](#)

Note that not all of the items listed may be relevant for all appliances.

Modify the TCP/IP Settings

If you set a static IP address and/or DNS server for the appliance and do not wish to preserve those settings, do the following:

Note: This step is optional.

1. Open the Windows **Control Panel**.
2. Click the link for **View network status and tasks** under the **Network and Internet** link.
3. Click the **Local Area Connection** (the Local Area Connection that is connected to the internet) link.
4. Click the **Properties** button.
5. Select **Internet Protocol Version 4 (TCP/IPv4)** and click the **Properties** button.
6. Verify that both **Obtain an IP address automatically** and **Obtain DNS server address automatically** options are selected and click the **OK** button.
7. Click the **Close** button in both the **Local Area Connection Properties** dialog and the **Local Area Connection Status** dialog.
8. Close the Network and Sharing Center window.

Verify Optional Configurations Still Exist

The Windows appliance we created in this chapter will be used to create other appliances later in this document, including an IIS appliance and a MS SQL Server appliance, which both have separate MSI files that need to be installed. When those MSI's are installed they will re-create the following directories and all of their contents:

- C:\appliance
- C:\cygwin

Any changes that were previously made to these directories will be lost. If you made changes according to either of the sections below, you will need to perform the steps again:

- [Disabling Auto Host Name Generation](#)

- [Disable Random Password Generation](#)

Reset the administrator password

Resetting the Windows Administrator's password is an optional step that can be used to encourage end users to set their own Windows Administrator account's password for their instance of the appliance. To reset the password do the following:

Note: Before you change the Windows Administrator account's password be sure to take into account the affect on other applications that may use this password. For example, some applications may be configured to use the Windows Administrator credentials. If this is true, it's possible you may need to update the credentials.

1. Open an SSH session into the singleton by selecting the AppLogic application within the CA AppLogic GUI and clicking the **Login (ssh)** button within the toolbar.

2. Set a new complex password by executing the following command:

```
net user administrator <new_password>
```

For example, the password could be set to **pq398hpaowht0293j^LWOIFH9htfw9jfe**.

3. Exit the SSH session.

Clear the Remote Desktop History Cache

If a remote desktop connection was made within the singleton to another system, do the following to clear the Remote Desktop History Cache:

1. Open the Windows Control Panel.
2. Click on the **Appearance** Link.
3. Click on the **Show hidden files and folders** link under **Folder Options**.
4. Select the option **Show hidden files, folders, and drives** in the **Folder Options** dialog, and click **OK**.
5. Open Windows Explorer and navigate to **C:\Users\Administrator\My Documents**.
6. Delete the **Default.rdp** file.
7. Click **Start > Run...**
8. Type **regedit** in the **Run** dialog, and click **OK**.
9. Navigate to **HKEY_CURRENT_USER\Software\Microsoft\Terminal Server Client\Default**
10. Delete all String Values beginning with **MRU**.
11. Close the registry.
12. Perform steps 1-4 again to select the option **Don't show hidden files, folders, or drives**, and clicking **OK** when done.
13. Close the Windows Control Panel.

Clear Windows Explorer Search Bar

To delete Windows Explorer search history remove the values from the following registry key:

```
HKCU | Software | Microsoft | windows | CurrentVersion | Explorer |  
wordwheelQuery
```

Clear the Bash History Cache

Whenever Cygwin is run, it opens a Bash Shell. When you execute a command in that Bash shell, chances are those commands will be saved, depending on how you exited the shell. To delete the Bash history, perform the steps below:

1. With the application started, SSH into the appliance by selecting the application in the CA AppLogic GUI or by opening the Application Editor by clicking on the application and then selecting the singleton to login to, and clicking the **Login (ssh)** button in the toolbar.
2. Execute the following commands:

```
rm -f ~/.bash_history  
history -c
```

3. Close the SSH session.

Collapse all Nodes in the Registry

Whenever the registry is opened, it remembers the last node that was opened. Open a Command Prompt and execute **regedit** to open the Registry Editor, and collapse all of the nodes. Close the **Registry Editor** window when done with the **Computer** node selected in the Registry Editor.

Collapse all Nodes in Server Manager

Whenever Server Manager is used, it also remembers which nodes you selected last, so when you close it, and open it up again, it will display the information of the last node selected. Click the **Server Manager** button on the desktop taskbar, and collapse all nodes. Before you exit, make sure that **Server Manager**, the root node, is selected. Now, when you open Server Manager next it will look like it did when you first installed the Operating System.

Delete Files/Logs

Depending on what you installed on the appliance you may wish to delete certain files, such as log files. Carefully review all software installed on the appliance and determine if any files need to be deleted. Examples include:

- C:\appliance\passwd.stamp

Deleting this file tells AppLogic to generate a new Windows Administrator password on the first boot of a new instance.

- C:\Users\Administrator\Desktop\Internet Explorer Troubleshooting

This gets installed if Internet Explorer 9 is installed.

- Log files under C:\appliance\init\logs, C:\cygwin\var\log, and C:\cygwin\var\log\appliance directories

Note: The C:\appliance\init\logs directory is used to hold the log files generated by the licensing scripts documented in Appendix A. Depending on how you choose to manage licensing and activation for your appliance this directory may not exist. The rest of the logs are generated whenever the appliance restarts.

- Browsing history in Internet Explorer:

To do this:

1. Open Internet Explorer

Note: Do not set up Internet Explorer now.

2. Click on **Tools**, and click **Internet options**.
3. In the **Internet Options** dialog, on the **General** tab, locate the **Browsing History** section and click the **Delete...** button to display the **Delete Browsing History** dialog.
4. Uncheck the **Preserve Favorites website data** checkbox.
5. Verify the rest of checkboxes are checked, and click **Delete**.
6. Click the **Delete...** button again, and check the **Preserve Favorites website data** checkbox again, and click the **Delete** button.
7. Check the **Delete browsing history on exit** checkbox, and click the **OK** button.
8. Close Internet Explorer

Clear entries in Event Viewer

To clear entries from the Event Viewer, do the following:

1. Click on **Start**, and click Administrative Tools.
2. Click Event Viewer.
3. Expand the Windows Logs section on the left hand pane.
4. Right click on each item and select **Clear Logs**

Clear the Start Menu Cache

Whenever you execute commands through the Windows **Start, Run** menu those commands are stored in memory. Also, when programs are executed, the programs are stored in the Start Menu as well. Perform the following steps to clear the Start Menu Cache:

1. On the Desktop, right-click the taskbar and click **Properties** in the context menu.

2. Click the **Start Menu** tab in the **Taskbar and Start Menu Properties** dialog.
3. Uncheck the following checkboxes in the **Privacy** section, and click the **Apply** button:
 - Store and display recently opened programs in the Start menu
 - Store and display recently opened items in the Start menu and the taskbar
4. Check those same checkboxes again, and click the **OK** button.

Empty the Recycle Bin

The final clean up step is to empty the Recycle bin within the Singleton. Close the graphical console when you are done.

Step 9: Add the Singleton to a Catalog

This step is optional; however, adding a singleton to a catalog will transform the singleton into an appliance that can be instantiated on demand in any application. Since this is the intention of our example, do the following:

1. Stop the AppLogic application by right-clicking the application in the CA AppLogic GUI, and clicking **Stop** in the context menu.
2. Move the singleton into the desired catalog. To do this:
 - a. Open the Application Editor by clicking the application in the CA AppLogic GUI.
 - b. Drag and drop the singleton onto the desired catalog (for example, system_ms).

Note 1: If the category you assigned this appliance to within the boundary of the appliance does not exist within the catalog, it will be created for you.

Note 2: You must assign yourself full access rights to the catalog before you can move the singleton. See the "[catalog modify_acl \(Catalog Management\)--Replace Portion of Global Catalog ACL](#)" section in the *CA AppLogic Command Line Shell Reference Guide* for more information.

Note 3: It is recommended that you add comments to the Notes tabs to track changes made to the appliance. The Notes tab includes a free form text field and can be accessed by right clicking on the singleton and selecting Modify Boundary.

The WIN0864S appliance is now ready for use.

In the [Preparation Work: Copy and Import the Files](#) section we created the system_ms catalog but you can also create a new global catalog for the appliance by doing the following:

1. Close the Application Editor.
2. Open the Grid Shell by clicking the **Grid Shell** button in the toolbar of the CA AppLogic GUI.
3. Execute the following command:

```
cat create /<name of catalog>
```

4. Open the Application Editor by clicking on the application within the CA AppLogic GUI.
The new catalog should be visible.

What if the move to catalog fails?

If the move to the catalog fails with an error message indicating that the common volumes of an appliance must be read-only and shared, do the following:

1. Open the class editor by right-clicking the singleton and click **Modify Boundary** in the context menu.
2. Click the **Volumes** tab and verify that all volumes of type **Common** (usually called `usr`) are marked as **read-only** and **shared**.

This allows CA AppLogic to share the common volume between all instances of the appliance. If you do not want to allow this, change the volume type to **instantiable**.

3. Close the class editor and try to move the appliance to the catalog again.

Step 10: Verify that the Appliance Starts

Now that the appliance has been created, we want to do a quick verification by starting the CA AppLogic application, `win08_install`, and verify the application starts.

Right-click the application in the CA AppLogic GUI, and click **Start** in the context menu. Verify the application starts.

What if the Appliance won't start?

The most likely reason that an appliance fails to start is because it has not started the VM agent. In the unlikely event this occurs you should consider restarting the appliance creation process to attempt to reproduce the error. If you are a system administrator or if you are familiar with advanced VM agent configuration you could also try some of the tactics listed below.

First, execute the following command in the Grid Shell to view the logs:

```
3t log list
```

If the VM agent does not start you will see that the appliance takes a long time start - only to time out and fail. To fix this, ensure that the appliance runs the `/appliance/vma_load.sh` script when it is done booting. All templates from the global catalog have this script and start it by default.

- Start the application with `--debug` flag on the `app start` command; this will make sure that CA AppLogic will leave the appliance in start-failed state after the boot timeout has expired.

For example:

```
app start win08_install --debug.
```

- Try to SSH into the appliance using the `comp ssh` command.

For example:

```
comp ssh win08_install:main.WIN0864S
```

If you can SSH into the appliance, navigate to the `/cygwin/tmp` folder and create a file called “`applogic_debug`” in the `tmp` folder. When you restart the appliance in debug mode this file will be populated with messages that may help with diagnosing the issue.

You can also navigate to the `/cygwin/var/log` folder to find any additional log files. At some point the VM agent needs to be loaded. When the boot completes `vme` should be executed with “`started_ok`”. For example `vme id=started_ok`.

If you cannot even SSH into the appliance, it is likely that the volume or its boot configuration has been corrupted. To resolve this you can stop the application, manage the volume and inspect it, or start from the template again. If the problem recurs, contact Technical Support.

What if the Appliance Failed to Start For Another Reason Besides a Timeout?

If the appliance start has not timed out, shut down the OS from within the graphical console.

This will cause the `appstart --debug` command to fail. Then, execute `app stop win08_install`. Otherwise, execute the `3t app stop win08_install` command in the SSH shell followed by shutdown within the graphical console.

Creating a Windows 2008 Filer Appliance

The Windows Filer is used to perform file-system level operations over Windows NTFS volumes. If you do not have a functional Windows Filer you cannot use native Windows file formats (such as `ntfs`, `fat`, `fat32`, etc.). It supports the following modes of operation:

- `format`: Formats the volume to the specified filesystem (e.g., execute `format`).
- `fscopy`: Performs a block-level copy from one volume to another followed by, as required, partition extend.
- `fsck`: Checks the file system on the volume.
- `fsrepair`: Checks and repairs the file system on the volume
- `manual`: Provides user-level access to the volume through both a Web GUI and root shell (through SSH)

Filesystem level operations over ntfs08 volumes are provided by the Sys_Filer_Window08 template application and the /system_ms:Filer_Windows08 appliance. Both of these appliances were imported from the CA AppLogic software version directory as part of the appliance creation (see Preparation Work: Copy and Import the Files step in the previous chapter for more details). However, before we can use the /system_ms:Filer_Windows08 appliance we need to first replace its empty boot volume with a working boot volume of the same operating system. To do this we are going to create a Windows Server 2008 Standard Edition Core Install 32-bit (either sp1 or sp2) appliance (because this is the same operating system that was used to build the /system_ms:Filer_Windows08 appliance) and replace the empty boot volume within the /system_ms:Filer_Windows08 appliance with the boot volume of this new Windows appliance.

Note: Filer only installs on the 32bit English version of the Win2003/2008 OS.

There are 8 steps to creating a Windows filer appliance:

- Step 1: Create the Base Windows 2008 Appliance
- Step 2: Configure Windows
- Step 3: Set the Singleton to be Fully Managed
- Step 4: Complete Windows Configuration
- Step 5: Modify Boundary and Instance Settings
- Step 6: Add the Singleton to a Catalog
- Step 7: Create a Windows Server 2008 Filer Appliance
- Step 8: Verify the Filer

Following is a walkthrough of these steps. For more information consult the RefWindowsInstallFiler document which is available from the following link:

http://doc.3tera.com/AppLogic30/en/Developer_Guide/WinInstallRef2008FilerInstallRef_1.html

Note that, although the steps appear similar to those we have just completed, there are subtle but important differences between the creation of these appliances.

Step 1: Create a Windows Server 2008 32-bit Appliance

The Filer_Windows08 appliance is based on Windows Server 2008 Standard Edition Core Install 32-bit (either SP1 or SP2). We are using the Core Install to reduce the size of the filer appliance boot volume.

The first step in creating this appliance is to create a base server class using the same OS as follows:

1. Copy a Windows 2008 Server 32-bit ISO image to the impex volume of the grid controller.

Note: You can use either the SP1 or SP2 ISO image. For the purposes of this example we are using the ISO image en_windows_server_2008_with_sp2_x86_dvd_342333.iso that was downloaded from Microsoft's MSDN site.

2. Open a SSH(using PuTTY for example) session to the grid controller
3. Execute the following command to use the iso2class utility to begin an installation of Windows Server 2008 Standard Edition Core Install 32-bit:

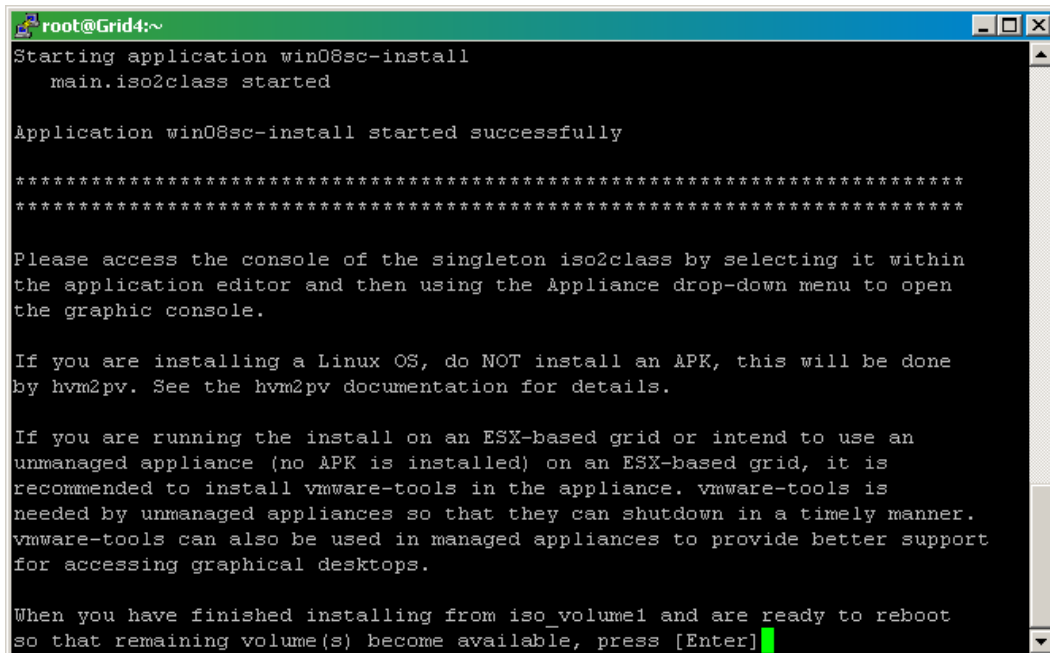
```
3t util iso2class app_name=win08sc-install  
iso_volume1=en_windows_server_2008_with_sp2_x86_dvd_342333.iso install_size=6G  
console_type=graphic virt_options=acpi=1 cpu=1 mem=1G.
```

Notes: The value specified for the Install_size parameter will depend on the size of the operating system installation and the disk space requirements of the deployment environment. Although we were successful using an install_size value of 6GB in our environment you may want to review Microsoft's minimum size recommendations to determine a value that is suitable for your environment and intent..

This command creates an application named "win08sc-install" with a singleton named "iso2class". The iso2class utility starts the win08sc-install application, booting the singleton iso2class from the specified ISO image.

4. When prompted within the PuTTY session to select an operating system type **9** for **Microsoft Windows Server 2008 (32-bit)**.

DO NOT press Enter when prompted after the iso_volume1 has finished installing. Continue on with the next step.



```
root@Grid4:~  
Starting application win08sc-install  
main.iso2class started  
  
Application win08sc-install started successfully  
  
*****  
*****  
  
Please access the console of the singleton iso2class by selecting it within  
the application editor and then using the Appliance drop-down menu to open  
the graphic console.  
  
If you are installing a Linux OS, do NOT install an APK, this will be done  
by hvm2pv. See the hvm2pv documentation for details.  
  
If you are running the install on an ESX-based grid or intend to use an  
unmanaged appliance (no APK is installed) on an ESX-based grid, it is  
recommended to install vmware-tools in the appliance. vmware-tools is  
needed by unmanaged appliances so that they can shutdown in a timely manner.  
vmware-tools can also be used in managed appliances to provide better support  
for accessing graphical desktops.  
  
When you have finished installing from iso_volume1 and are ready to reboot  
so that remaining volume(s) become available, press [Enter]
```

5. Use one of the following methods to access the graphic console of the singleton:

- Select the **win08sc-install** application within the application list in the CA AppLogic GUI and click on the icon for **Login (graphic)** found in the toolbar.
- Click the **win08sc-install application** in the CA AppLogic GUI to open the Application Editor and select the singleton iso2class. Then select **Login(graphic)** from the **Appliance** drop down menu.

Note: If you receive a “permission denied” message in response to these actions, see [What Happens If I Receive a Permission Denied Message](#) for a list of steps to take.

6. Respond to the prompt(s) regarding source and authenticity according to your current security policies and settings.

After the graphical console appears, the Windows 2008 Server installation interface should be visible.

7. Install the OS by following the prompts.

Note: When asked to select the operating system, choose Windows Server 2008 Standard (Server Core Installation).

The singleton will reboot during installation, rendering the graphical console inoperative. When this happens, simply close and re-open the graphical console by clicking the **Login (graphic)** button in the toolbar of the CA AppLogic GUI.

When the installation is complete, you will need to set the administrator’s password. To do this:

1. Move the mouse to the top of the graphical console, and click the **Ctrl-Alt-Delete** button.
2. Click **Other User**
3. Specify **administrator** in the *User name* field, leave the *Password* field blank and click **Login**.
4. When prompted to change the password, click **OK**.
5. Provide a new password in the *New Password* and *Confirm* password fields and press **<enter>**.
6. Click **OK** button when prompted that the password has been changed.

Step 2: Configure Windows

The next step is to configure the Windows installation. If you are creating Windows appliance on an **ESX grid**, however, you will first need to install VMware tools as follows:

1. Stop the application by right-clicking the application in the CA AppLogic GUI, and clicking **Stop** in the context menu.
2. In grid shell execute the following command:


```
3t vo1 copy _GLOBAL_RO:vmware_tools_windows win08sc-install:vmware_tools_windows
```
3. Open the Application Editor by clicking the application in the CA AppLogic GUI.
4. Right-click on the singleton appliance **iso2class** and select **User Volumes**

5. Select volume **vmware_tools_windows** for placeholder **iso_volume1**
6. Save and start the application.
7. Login through the graphical console
8. Press **Ctrl+Alt+Del** button on center top of the graphical console window and login with the Windows Administrator account.
9. Click inside the graphical console window anywhere
10. Within the Command Prompt that appears, execute the following to start the VMware tools installation silently:

```
D:\setup.exe /S /v /qn
```

The installation will reboot the appliance automatically. Wait patiently.

11. After reboot you have to reopen the graphical console by selecting the application in the CA AppLogic GUI and clicking the **Login (graphic)** button in the toolbar.
12. Once in, login to the appliance with the appropriate Windows credentials.

The first configuration change is made to the **default page file size**. Change this to 16M by doing the following:

1. With the desktop and the Command Prompt visible, set the page file size to 16M by executing the following from the command prompt:

```
wmic computersystem where name="%COMPUTERNAME%" set AutomaticManagedPagefile=False
wmic pagefileset where name="C:\\pagefile.sys" set InitialSize=16,MaximumSize=16
```

2. Reboot by issuing the following command:

```
shutdown -r -t 0
```

3. After a minute, re-open the graphical console by clicking the **Login (graphic)** button within the toolbar of the CA AppLogic GUI, and login to the Windows desktop with the appropriate credentials.

Next, **disable power management hibernation support** by issuing the following command:

```
powercfg -h off
```

Next, execute the following command:

```
dir
```

There should be approximately 4.8GB of free space on the disk.

Next, set the **default Windows disk device timeout value** to 60 seconds by doing the following:

1. Execute **regedit** to open the registry editor.
2. Navigate to the following key:

```
HKEY_LOCAL_MACHINE/System/CurrentControlSet/Services/disk
```


3. Right-click on **disk** in the left pane and click **New > DWORD (32-bit) Value**.
4. Type **TimeOutValue** and press the <enter> key.
5. Right-click on **TimeOutValue** and click **Modify...**
6. Set the **Base** to **Decimal** and enter **60** for the value data. Click **OK**.

The next step, **activating Windows**, is optional. Depending on how licensing will be managed for your appliance you may elect not to activate Windows at this point – leaving that step to the client who eventually provisions an application that uses this appliance. If so, skip to [Step 3: Set the Appliance to be Fully Managed](#).

Note: See [Licensing: Microsoft Windows Server](#) for more details.

1. Configure external network access by running the following commands:

```
netsh interface ipv4 set address name="Local Area Connection" source=static
address=X.X.X.X mask=X.X.X.X gateway=X.X.X.X (use valid network settings)
```

```
netsh interface ipv4 add dnsserver name="Local Area Connection" X.X.X.X index=Y
(where X.X.X.X is the IP address of the DNS Server and Y is its place in the list,
for example 1)
```

2. Verify external network access (for example, ping www.ca.com)
3. Update the Windows 2008 Server product key and activate Windows:
4. Update the product key by issuing the following command:

```
s1mgr.vbs -ipk xxxxx-xxxxx-xxxxx-xxxxx-xxxxx
```

Note: Ensure that you have provided the appropriate product key; wait until a message is displayed saying the product key was successfully installed before proceeding.

5. Activate Windows by performing the following command:

```
s1mgr.vbs -ato
```

Note: Two popup messages will be shown; wait until the Product was activated successfully message is displayed.

6. Verify activation by executing the following command:

```
s1mgr.vbs -xpr
```

Note: You should see a pop-up indicating Windows is activated; if not, repeat the above steps.

Finally, configure the Windows Server to obtain the IP address, and the DNS server automatically by executing the following commands:

```
netsh interface ipv4 set address name="Local Area Connection" source=dhcp
netsh interface ipv4 set dns name="Local Area Connection" source=dhcp
```

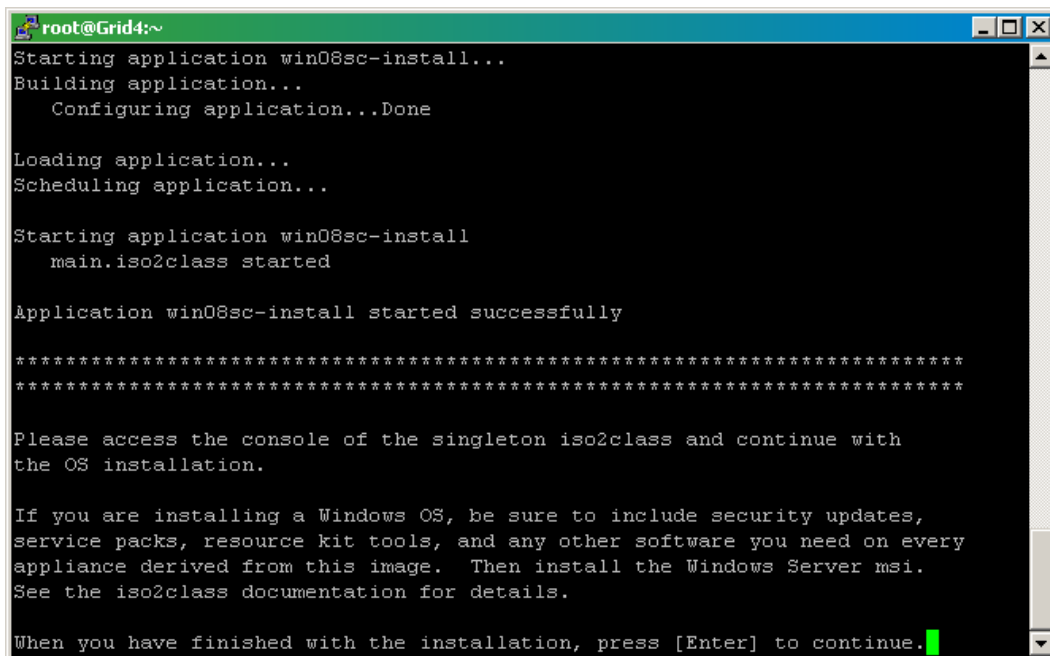
Step 3: Set the Singleton to be Fully Managed

Next we configure the singleton to be “fully managed” by CA AppLogic by installing the Windows Server MSI, which is located on one of the volumes of the singleton. To do this:

1. Shutdown the OS and continue the iso2class install as follows:
 - a. Execute the following command in the command prompt:
`shutdown -s -t 0`
 - b. Wait 5 seconds and then, return the SSH session that was open from the section [Step 1: Create a Windows Server 2008 32-bit Appliance](#) (step 4).
 - c. Press the **<enter>** key within the SSH session indicating that you have finished installing from iso_volume1.
 - d. When asked for confirmation, type **Y** and press the **<enter>** key again.

Note: A second volume will now be available to the singleton.

DO NOT press Enter or otherwise close the window within the SSH session when prompted at the completion of the install.. For now, continue to the next step.



```
root@Grid4:~
Starting application win08sc-install...
Building application...
  Configuring application...Done
Loading application...
Scheduling application...

Starting application win08sc-install
  main.iso2class started

Application win08sc-install started successfully

*****
*****

Please access the console of the singleton iso2class and continue with
the OS installation.

If you are installing a Windows OS, be sure to include security updates,
service packs, resource kit tools, and any other software you need on every
appliance derived from this image. Then install the Windows Server msi.
See the iso2class documentation for details.

When you have finished with the installation, press [Enter] to continue.
```

2. After a minute re-open the graphical console by selecting the application within the CA AppLogic GUI and clicking the **Login (graphic)** button within the toolbar.
3. Login to the Windows desktop using the appropriate Windows credentials.
4. Execute the following from a command prompt to navigate to the E drive:
`E:`
5. Execute the following command to install the Windows Server MSI
`server_windows-<version>.msi`

Note: For the purposes of this example we have used version 2.0.2-1.

6. Follow the prompts to complete the basic installation of the Server_windows MSI:
7. Execute the following command within the command prompt:

```
shutdown -s -t 0
```
8. Wait 5 seconds, and then return to the previously opened SSH session (see Step 2).
9. Within the SSH session running the iso2class utility, press the <enter> key and then type **Y** to confirm installation has finished, and press the <enter> key again.
10. When prompted in the SSH session, type **1** to specify that **Windows msi/APK is installed: the appliance is fully managed**, and press the <enter> key.

This iso2class will stop the application, change the singleton boundary to that of a generic server appliance, and re-start the application (the appliance will enter maintenance mode during the restart).

```
root@Grid4:~
When you have finished with the installation, press [Enter] to continue.

Are you certain? Specify 'Y' to continue and then press [Enter]: Y

*****
*****

Stopping application win08sc-install
main.iso2class stopped

Releasing application resources...
Application win08sc-install stopped

*****
*****

Before continuing with the final modification of the singleton class
descriptor, please specify how CA 3Tera AppLogic should manage the resulting
appliance:
  1. Windows msi/APK is installed: the appliance is fully managed
  2. Windows msi/APK is NOT installed: the appliance is unmanaged

Please specify a number 1-2 and then press [Enter]: █
```

11. Verify that you see the message indicating that iso2class has completed its operation.

```
root@Grid4:~
Starting application win08sc-install...
Building application...
  Configuring application...Done

Loading application...
Scheduling application...

Starting application win08sc-install
  main.iso2class entered maintenance state
  main.iso2class started

Application win08sc-install started successfully

*****
*****
iso2class has completed its operation. The resulting singleton iso2class
in application win08sc-install uses HVM virtualization (hardware emulation).

For more information, see the on-line reference at:
http://doc.3tera.com/AppLogic30/RefIso2Class.html

APPLOGIC RESTRICTED AREA
[root@Grid4 ~]#
```

Step 4: Complete Windows Configuration

We will now complete the Windows configurations by performing the following steps:

1. Login through the graphical console by selecting the application within the CA AppLogic GUI and clicking the **Login (graphic)** button within the toolbar.
2. Login to Windows using the appropriate credentials.
3. In the Command Prompt change the computer name by issuing the following command:

```
wmic computersystem where name="%COMPUTERNAME%" rename name="WIN08SC"
```

4. Open an SSH session into the singleton by selecting the application within the CA AppLogic GUI and clicking the **Login (ssh)** button within the toolbar,
5. Execute the following command:

```
rm -f /appliance/passwd.stamp
```

6. Using the SSH session, execute the following command to set a complex Administrator password:

```
net user administrator <new_password>
```

For example, the password could be set to pq398hpaowht0293j^LWOIFH9htfw9jfe

This is done to encourage others to reset the Windows Administrator account's password before logging into the appliance.

7. Execute the following commands to clear the password from the Bash history:

```
rm -f ~/.bash_history
history -c
```

8. Exit the SSH session by closing the SSH window.

Step 5: Modify Boundary and Instance Settings

In this step we will be modifying the boundary of the singleton to prepare it to become an appliance. To do this:

1. Close the graphical console.
2. Stop the application in the CA AppLogic UI by right-clicking the application and selecting **Stop** from the context menu.
3. Click on the application in the AppLogic GUI to open it in the AppLogic editor.
4. Right-click on the singleton and select **Attributes**.
5. Change the instance name to **WIN08SC** in the Instance Settings window, and click **OK**.
6. Right-click on the singleton and select **Modify Boundary**:
7. Make the following changes to the boundary:

On the **General** tab:

- Change the class name to **WIN08SC**.
- Change the Category to **Operating Systems**.
- Provide a description. For example: **Windows Server Appliance - based on Windows Server 2008 Standard Edition Core Install SP2 32-bit**.
- Change the Color to **Blue**.
- Optionally, set the size as necessary.
- Set the OS Icon to **Windows**.

On the **Resources** tab, change the following settings:

- CPU (num)
 - Min: 0.25
 - Max: 4
 - Default: 1
- Memory (bytes)
 - Min: 512M
 - Max: 4G
 - Default: 1G
- **Bandwidth (bits/sec)**
 - Min: 1M
 - Max: 2G

- Default: 100M
8. Click the **OK** button.
 9. When prompted to save the application, click the **OK** button.

Step 6: Add the Singleton to a Catalog

Next we transform this singleton into an appliance by moving it into the /system_ms catalog. To do this:

1. Select the system_ms catalog from the drop-down menu in the upper left-hand corner of the Application Editor.
2. Drag and drop the singleton into the catalog.

Note: You must have full access rights to the catalog before you can move the singleton into it. For more information on catalog access rights see [catalog modify acl \(Catalog Management\)-- Replace Portion of Global Catalog ACL](#) in the *Command Line Shell Reference Guide*.

Step 7: Create a Windows Server 2008 Filer Appliance

In this step we are going to create a new Windows Server 2008 filer appliance to replace the existing filer appliance in the system_ms catalog. We will then copy the boot volume from the WIN08SC appliance that we previously created to the new filer appliance.

To do this:

1. Right-click the **WIN08SC** appliance within the Application Editor for the **win08sc-install** application and select **Branch** from the context menu.
2. Right-click on the appliance in application editor and select **Modify Boundary** from the context menu.
3. Click the **Volumes** tab
4. Click **Add** to create a new volume and specify the following information:
 - Specify the Type to be **Placeholder**
 - Specify the Name to be **apk_volume**
5. Click **Next**.
6. Click in the field under the **Mount on** column, for the **apk_volume**, and type **D:**
7. Click **OK** in the Class Definition window.
8. Right-click the singleton and select **User Volumes** from the context menu.
9. Click in the field under the column **App Volume** for the **apk_volume**, and select **apk_volume** from the drop-down list.

Note: If this volume is not present then run the following command to copy the volume to the application:

```
vol copy _GLOBAL_R0:apk_windows win08sc-install:apk_volume
```

Once there, then perform this step again.

10. Click the **OK** button in the Instance Settings window.
11. Click the **Save** button within the toolbar of the Application Editor
12. Close the Application Editor.
13. Right-click the application in the CA AppLogic GUI and click **Start**.
14. Once the application to finish starting click the **Login (ssh)** button within the tool bar of the CA AppLogic GUI.
15. Execute the following command to set the Administrator password:

```
net user administrator <new_password>
```

16. Click the **Login (graphic)** button in the toolbar of the CA AppLogic GUI to open the graphical console, and login to Windows with the appropriate credentials.
17. In the command prompt, execute the following command to install the Windows Filer MSI:

```
D:\Filer_windows-<version>.msi
```

Note: The version that was used in this example is 3.0.5-1

18. Close the graphical console.
19. Within the SSH session:
 - a. Set a complex Administrator password by executing the following command so that nobody can login to the appliance as the Administrator without first setting a new password:

```
net user administrator <new password>
```

For example the password could be asdfasdjJJDFSDF99^HHR
 - b. Execute **rm -f ~/.bash_history** to clear the password from the Bash history.
 - c. Exit the SSH session by closing the window.

20. Right-click the application within the CA AppLogic GUI and select **Stop** from the context menu.
21. Open an external SSH session(using PuTTY for example) to the grid and execute the following command:

```
3t class unlock /system_ms:Filer_windows08
```

This unlocks the appliances (which was locked when it was imported), enabling it to be modified.

Note: This action requires grid administrator access to the grid.

22. Click the **New application** button within the toolbar of the CA AppLogic GUI and name the application **win08filer-install**.

23. Edit the new win08filer application by clicking on it and dragging the appliance **Filer_Windows08** onto the canvas from the **/system_ms catalog**.
Note: This appliance was imported as part of the /system_ms catalog in the section [Preparation Work: Copy and Import the Files](#).
24. Right-click the **Filer_Windows08** appliance and select **Branch** from the context menu.
25. Save the application by clicking the **Save** button in the toolbar of the Application Editor.
26. Right-click the singleton and click **Modify Boundary**.
27. On the **General** tab, set the Documentation URL to <http://doc.3tera.net/AppLogic30/CatFilerWindows08.html> and click the **OK** button.
28. Save the application by clicking the **Save** button within the toolbar of the Application Editor.
29. Execute the following commands within a SSH session(using PuTTY for example):

```
3t vol destroy win08filer-install:Filer_windows08.boot --force
3t vol copy win08sc-install:WIN08SC.boot win08filer-install:Filer_windows08.boot
3t class destroy /system_ms:Filer_Windows08 --force
```

Notes: These commands will do the following:

- Destroy the empty boot volume of the existing Filer_Windows08 appliance.
 - Copy the boot volume of the WIN08SC Windows appliance to the Filer_Windows08 singleton within the win08filer-install application.
 - Destroy the original Filer_Windows08 appliance within the /system_ms catalog.
30. Within the Application Editor for the **win08filer-install** application, drag the Filer_Windows08 singleton into the **/system_ms catalog**.
This creates the new Filer_Windows08 appliance in the catalog.
 31. Execute the following command within the SSH session (PuTTY for example) to lock the new Windows Filer appliance so no one can modify it:

```
3t class lock /system_ms:Filer_Windows08
```

Note: This requires administrator access to the grid.

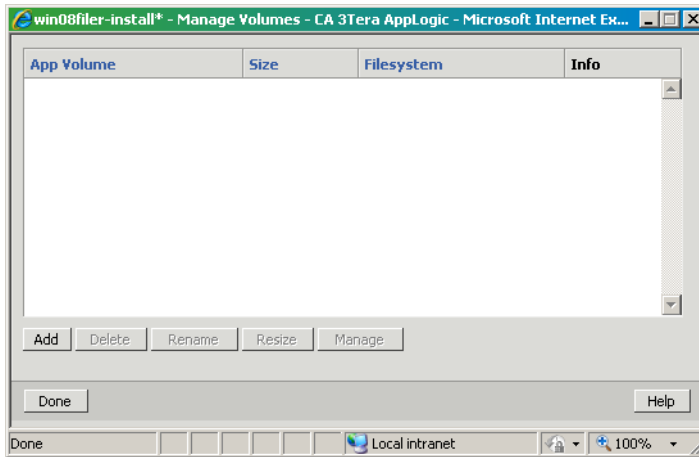
The /system_ms:Filer_Windows08 appliance class is now complete and ready for use.

Step 8: Verify the Filer

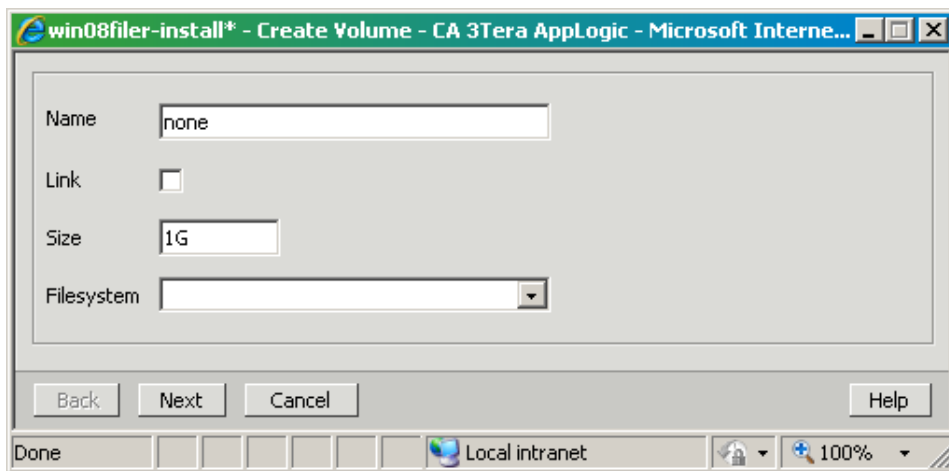
The final step is to verify that the newly created Filer appliance is complete and functional. To do this, create or open an application using the Filer, then add an NTFS volume to it.

For example:

1. Click the **win08filer-install** application within the CA AppLogic GUI to open the Application Editor.
2. On the toolbar click the **Application** menu, and click **Manage Volumes** in the context menu.

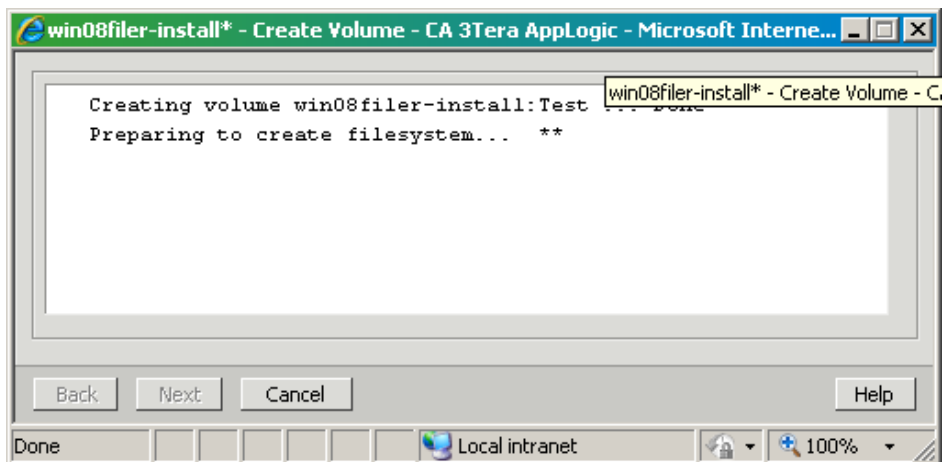


3. Click **Add**.



4. Provide a value for **Name** and **Size**.
Note: The smaller the volume, the more quickly it will be created.
5. Choose the **ntfs08** option from the **Filesystem** drop-down list and click **Next**.

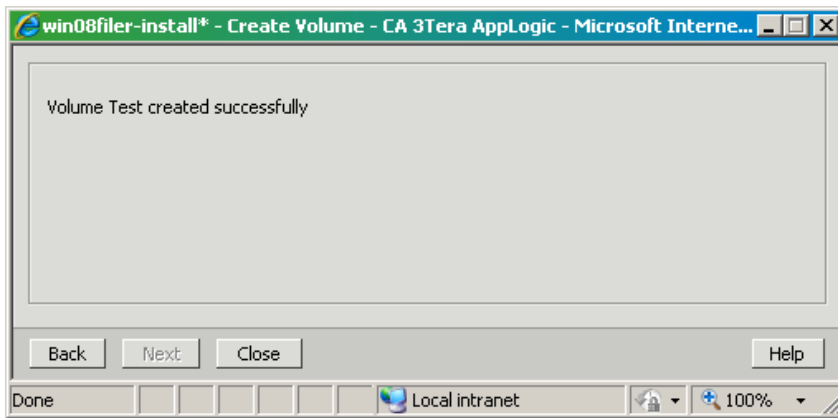
The filer processes your request.



During processing, AppLogic will create an additional application in your application list which will run, create the file system, and then disappear...

Dashboard	Applications	Logs	Support
		5.2.0 (v5.2.0-10)	
Sys_Filer_Linux (template)	Stopped	Linux Filer Application (v4.0.2-1)	0.05 512.00M 1.00M
Sys_Filer_Solaris (template)	Stopped	Solaris Filer Application (v3.0.4-1)	0.05 512.00M 1.00M
Sys_Filer_Windows08 (template)	Stopped	Windows08 Filer Application (v3.0.1-1)	0.25 512.00M 1.00M
TWiki_113 (template)	Stopped	Twiki 4.3.2 collaboration platform (v4.3.2-1)	1.25 1.25G 900.00M
VDS_CentOS55_r3 (template)	Stopped	Virtual Dedicated Server - Based on CentOS 5.5 (v1.0.2-1)	0.25 256.00M 1.00M
VDS_OSOL_r13 (template)	Stopped	Virtual Dedicated Server - Based on OpenSolaris build 2008.11 32-bit (v1.1.2-1)	0.50 512.00M 1.00M
VD564_CentOS55_r3 (template)	Stopped	Virtual Dedicated Server - Based on 64-bit CentOS 5.5 (v1.0.2-1)	0.25 512.00M 1.00M
VD564_OSOL_r17 (template)	Stopped	Virtual Dedicated Server - Based on OpenSolaris build 2008.11 64-bit (v1.2.2-1)	0.50 512.00M 1.00M
WS_API_r7 (template)	Stopped	REST - based AppLogic Web Service API (v1.0.7-1 BETA)	1.10 1.66G 730.00M
WS_API_SAMPLE_r2 (template)	Stopped	AppLogic API Sample Application (v1.0.1-1)	1.05 1.94G 1.15G
Sys_Filer_win08filer-install-user-Test	Building	Windows08 Filer Application (v3.0.1-1)	0.25 512.00M 1.00M

...and complete. Here we can see that the test was successful:



Once the test volume has been successfully created, you can delete it through the Application, Manage Volumes option (see step one above).

Chapter 4: Creating a Microsoft SQL Server Appliance

At this point in the document we have already created the WIN0864S appliance and added it to the catalog. In this chapter we will demonstrate how to use the WIN0864S appliance to create a new Microsoft® SQL Server® 2008 R2 appliance.

Following is a list of steps we will follow in this chapter:

- [Step 1: Create an AppLogic Application for the MS SQL Server 2008 R2 Appliance](#)
- [Step 2: Modify WIN0864S to Install MS SQL Server](#)
- [Step 3: Prepare Windows to Install MS SQL Server](#)
- [Step 4: Install MS SQL Server](#)
- [Step 5: Install the MS SQL Server MSI](#)
- [Step 6: Script the Operations to License MS SQL Server Upon Startup](#)
Note: This step is optional.
- [Step 7: Prepare the MS SQL Server Singleton to be Added to a Catalog](#)
- [Step 8: Add the Singleton to a Catalog](#)
- [Step 9: Verify your MS SQL Server 2008 Appliance](#)

For more information on these procedures refer to the *CA AppLogic Appliance Developer Guide*, section SQL08X/WG/D/W/S/E_08: Microsoft SQL Server Database Appliances – Installation Reference which is available from the following link:

http://doc.3tera.com/AppLogic30/en/Developer_Guide/DistInstallSQL08X_1.html

Step 1: Create an AppLogic Application

Since MS SQL Server 2008 R2 must be installed on a Windows operating system, the first step in the process of creating an MS SQL Server appliance is to prepare a Windows appliance. For the purposes of our example we will use the WIN0864S appliance we created in the previous chapter. A new application will be created as the work area for building the new MS SQL Server appliance.

To create the application, do the following:

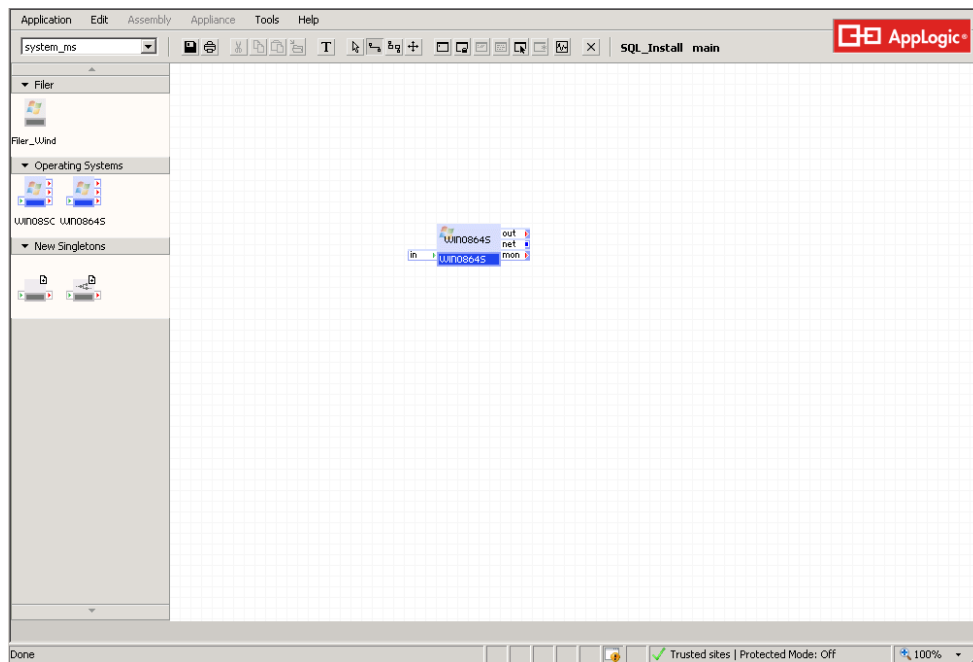
1. Upload the MS SQL 2008 R2 ISO image to the /vol/_impex directory on the AppLogic's grid controller.
2. Login to the CA AppLogic GUI dashboard, and click the **Applications** tab.

3. Click the **New application** button in the tool bar of the CA AppLogic GUI.
4. Provide a name for the MS SQL Server 2008 R2 application and click the **OK** button.

Step 2: Modify WIN0864S to Install MS SQL Server

The MS SQL Server appliance we are creating will use the WIN0864S appliance we already built; however, we need to make the following modifications to that appliance in order to install MS SQL Server on it:

1. Click on the new application to open the Application Editor.
2. Drag the **WIN0864S** appliance from the **/system_ms** catalog onto the editor.



3. Right click the appliance and select **Branch Class**.
4. Save the application when prompted to do so.
5. Click the **Close** button when done.
6. With the Application Editor opened, right-click the singleton, and select **Modify Boundary**.
7. Select the **Volumes** tab and add the following volume by clicking the **Add** button and specifying the following information:
 - **Type:** Placeholder
 - **Name:** data
8. Click **Next**
9. Click the **Mandatory** button under the **Options** column, for the **data** volume.
10. Click the field under the **Mount on** column for the **data** volume, and type **D:**.

11. Repeat steps 7-10 with the following information:

- **Type:** Placeholder
- **Name:** SQL_ISO
- **Mount on:** E:\

Mark the SQL_ISO volume as **Read-only** and **Share** (instead of Mandatory), under the **Options** column.

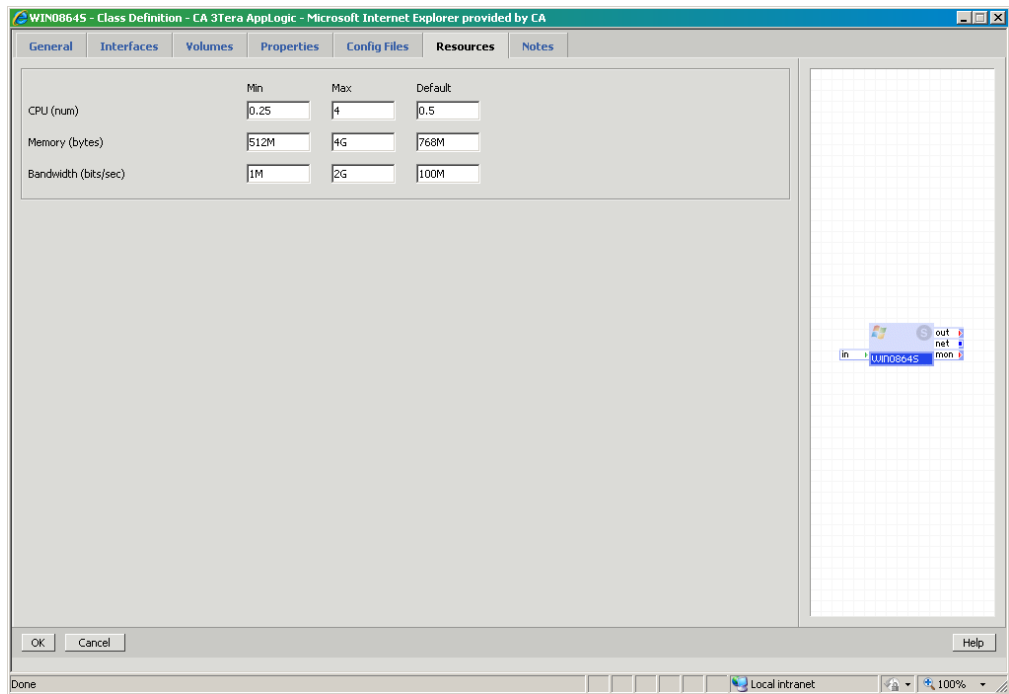
12. On the **Volumes** tab, select the **boot** volume and click **Resize**.

13. In the Resize Volume dialog, change the size to **22G**, and click **Next**.

Note: This step may take a while to complete.

14. Click **Close** when done.

15. Click the **Resources** tab:



16. Set the resource values for the singleton as follows:

Category	Minimum	Maximum	Default
CPU	0.25	32	1
Memory	1G	64G	2G
Bandwidth	1M	2G	100M

17. Click **OK** to save the changes.

18. Click the **Save** button in the toolbar of the Application Editor.
19. Click the **Grid Shell** button in the toolbar of the Application Editor to open a grid shell session on the AppLogic grid controller and execute the following command to import the MS SQL Server 2008 R2 ISO image into the application:

```
vol import <application_name>:SQL_ISO <ms_sql_server_image>.iso
```

For example:

```
vol import MS_SQL2008_Install:SQL_ISO  
en_sql_server_2008_r2_standard_x86_x64_ia64_dvd_521546.iso
```

20. Once the import is finished click the **Application** menu in the Application Editor tool bar and select **Manage Volumes** in the context menu.
21. Verify the **SQL_ISO** volume exists.
22. Add a new application user volume that will be used temporarily by clicking the **Add** button and specifying the following information. Once done, click **Next** in the Create Volume window:
 - Name: data
 - Size: 8G
 - Filesystem: ntfs08
23. Click the **Close** button in the Create Volume window when the operation completes.
24. Click the **Done** button within the Manage Volumes window.
25. Within the Application Editor, right-click the singleton, and select **Attributes** in the context menu.
26. Change the **Name** value to **SQL0864S**.
27. Click the **User Volumes** tab in the Instance Settings window.
28. Click in the field under the **App Volume** column, for the **data** volume, and select **data** from the drop-down list.
29. Repeat step 28 to set the **SQL_ISO** placeholder volume to use the **SQL_ISO** application volume.
30. Click **OK** in the Instance Settings window.
31. Click the **Save** button in the toolbar of the Application Editor and close the Application Editor.
32. Right-click the application in the CA AppLogic GUI and click **Start** in the context menu.

Step 3: Prepare Windows to Install MS SQL Server

Now that the singleton has been prepared to install MS SQL Server, the next step is to prepare Windows to install MS SQL Server. Perform the following steps:

1. Select the application in the CA AppLogic GUI, and click **Login (graphic)** button within the toolbar of the CA AppLogic GUI to login into the singleton using the graphic console.
2. When the console is displayed login to Windows using the appropriate credentials.

Note: If the [you did not disable random password generation](#), you will need to open an SSH shell by clicking the **Login (ssh)** button and execute the following command to set the administrator's password:

```
net user administrator <new_password>
```

3. Click **Start, Computer**.

If the CD Drive with the MS SQL Server image does not appear, perform the following steps to show the CD drive; otherwise, skip this step:

- a. Open the Server Manager by clicking the **Server Manager** icon on the task bar on the desktop.
 - b. Expand **Storage** and click **Disk Management**
 - c. Right-click the **CD-ROM 0** partition (this is the iso for MS SQL Server), and click **Change Drive Letter and Paths...** in the context menu.
 - d. Click the **Add...** button.
 - e. Verify that the letter **E** is selected in the drop-down menu and click **OK**.
 - f. Close Server Manager.
 - g. Verify in the Computer window that a second hard disk drive appears.
4. Enable the virtual memory for the singleton by performing the following steps:
 - a. Open the Windows Control Panel by clicking **Start > Control Panel**
 - b. Click the **System and Security** link.
 - c. Click the **System** link
 - d. Click the **Advanced system settings** link
 - e. Click the **Settings...** button, in the **Performance** section, of the **Advanced** tab, of the **System Properties** dialog.
 - f. Click the **Advanced** tab
 - g. Click the **Change...** button in the **Virtual memory** section.
 - h. For the C drive, verify that **No paging file** is selected, and click the **Set** button. When prompted to continue click **Yes**.
 - i. Select the D drive in the **Paging file size for each drive** listbox.

- j. Select the option **System managed size** and click the **Set** button.
 - k. Click the **OK** button in the **Virtual Memory** dialog. When notified that the computer needs to be restarted, click **OK**.
 - l. Click the **OK** button in the **Performance Options** dialog.
 - m. Leave the **System Properties** dialog open.
5. If the option to [automatically generate a hostname was disabled](#), then perform the following steps to manually set the computer name; otherwise, click the **OK** button in the **System Properties** dialog, and continue on with the next step:
 - a. Click the **Computer Name** tab within the **System Properties** dialog.
 - b. Click the **Change...** button.
 - c. Change the Computer name to **SQL0864S** and click the **OK** button.
 - d. When prompted to restart the computer, click **OK**.
 - e. Click the **Close** button in the **System Properties** dialog.
 6. Click the **Restart Later** button.
 7. Close the graphical console.
 8. Right-click the application within the CA AppLogic GUI and click **Restart** in the context menu to restart the application.

Can I change the node name after MS SQL Server is installed?

Since MS SQL Server associates the node name with the instance name, if you change the node name later it will be unable to find your database. To resolve this situation do the following:

1. Open an MS SQL Server Management Studio tool and open a new query.
2. Execute the following commands:

```
sp_helpserver (returns the current server name)
sp_dropserver current_name
sp_addserver new_name
sp_helpserver (to verify success)
```

What if there are multiple MS SQL Server appliances in the same application?

When you drag and drop an appliance instance onto the canvas of a new application, if CA AppLogic detects that there is already an existing instance of the same appliance, it will add an incremental suffix to the instance name of the new instance. For example, if the application already includes the instance SQL0864S1 when a second instance of the same appliance is dragged in, it will automatically be named SQL0864S2 –this will ensure that both appliances are unique. Since changing the instance name also changes the Windows hostname, MS SQL Server will also be affected so you will also need to [update the SQL Server with the new host name](#).

Step 4: Install MS SQL Server

The next step is to install Microsoft SQL Server 2008 R2 silently using a configuration file found in [Appendix B](#). Perform the following steps:

1. With the application selected in the CA AppLogic GUI, click the **Login (graphic)** button within the toolbar to open the graphical console.
2. Login to Windows using the appropriate credentials.
3. Using an SSH Shell (like PuTTY) to log into the singleton, create a text file called **D:\ConfigurationFile_x64_R2.ini (/cygdrive/d/ConfigurationFile_x64_R2.ini)**.
4. Copy and paste the contents from [Appendix B: Microsoft SQL Server Configuration File](#) into this new file. Save the file when done, and close the SSH session.

Note: Be sure to verify that you have copied the full contents of the file.

5. Open a command prompt within Windows singleton, and change the current directory by executing the following command

E:

6. Execute the installer for Microsoft SQL Server by executing the following command:

```
setup.exe /q /action=Install /configurationfile=D:\ConfigurationFile_x64_R2.ini  
/MEDIASOURCE=E:\ /sapwd="manager" /ASSYSADMINACCOUNTS="%COMPUTERNAME%\Administrator"  
/SQLSYSADMINACCOUNTS="%COMPUTERNAME%\Administrator" /TCPENABLED=1
```

Notes:

- MEDIASOURCE should point on the root of the SQL installation folder (in this case "E:")
 - You can specify an sa user password other than "manager" if desired.
 - If the installer notifies you that the .NET framework must be enabled, open Server Manager->Features, and enable the .NET framework 3.5.1 feature. Then, run the above installation command again.
7. Install the latest patches and service packs for Microsoft SQL Server.

Note: If you require external network access, you may have to enable the External Interface within the boundary of the singleton and provide a static ip address, network mask, gateway, and DNS IP address for the Local Area Connection adapter. To enable the external network access perform the following steps:

- a. Right-click the application in the CA AppLogic GUI, and click stop.
- b. Click the application to open the Application Editor.
- c. Right-click the singleton SQL0864S and click **Modify Boundary**.
- d. Click the **Interfaces** tab.
- e. Check the **External Interface** checkbox, and click **OK** button to save the changes.
- f. Click the **Save** button in the toolbar of the Application Editor.

- g. Right-click the application within the CA AppLogic GUI, and click **Start**.
- h. Select the application in the CA AppLogic GUI and click **Login (graphic)** button within the toolbar of the CA AppLogic GUI to login into the singleton using the graphic console.
- i. When the console is displayed login to Windows using the appropriate credentials.
- j. Open Server Manager by clicking the **Server Manager** button in the taskbar at the bottom of the desktop.
- k. With Server Manager (... selected, Click View Network Connections in the Computer Information section.
- l. Double-click Local Area Connection
- m. Click the **Properties** button.
- n. Select Internet Protocol Version 4 (TCP/IPv4) and click the Properties button.
- o. Select **Use the following IP address** and enter usable values for your grid (IP Address, netmask, gateway, DNS server). Click **OK** when done.
- p. Click the Close button in the Local Area Connection Properties dialog.
- q. Click the Close button in the Local Area Connection Status dialog
- r. Repeat Step 7 to install the latest patches.

Step 5: Install the MS SQL Server MSI

The MS SQL Server MSI copies all of the contents of the D drive (including the system database tables) and stores them in a zipped file. Later, after this singleton becomes an appliance, whenever an end user drags an instance of this new MS SQL Server appliance into their application, upon startup of the application, CA AppLogic will unzip the contents of those files and store them on the D drive. For all of the system databases, CA AppLogic will attach those databases so that Microsoft SQL Server will automatically start without any manual intervention.

This MS SQL Server MSI can be downloaded from the grid controller through the default interface of this singleton.

Note: When the MSI installs, it deletes the following folders, and all of their contents and removes any permissions that may have been set:

- C:\appliance
- C:\cygwin

Any changes made to those folders should be backed up now. For the purposes of this example, we had to backup the following folders and files:

- C:\appliance\init (and all of its contents)

This folder structure contains all of the scripts that were written for the licensing example. Both the folder and its contents must be backed up.

Note: C:\Cygwin\etc\sysconfig\applogic_appliance will not be backed up even though we made modifications to it for the licensing example in this document. This file will just be modified again in the section [Step 6: Optionally Script the Operations to License MS SQL Server Upon Startup](#).

To install the MS SQL Server MIS, do the following:

1. Locate the ip address of the grid controller (DHCP Server) by performing the following steps within the singleton:

- a. Open a command prompt and execute the following command:

```
ipconfig /all
```

- b. Review each of the **Ethernet adapter Local Area Connection X** sections within the command prompt, and look for the property **DHCP Server**. The Local Area Connection that contains this property is the default interface. Note the IP address of the DHCP server.

Note: In this example, Local Area Connection 3 contained the DHCP Server property but that may be different with each implementation.

2. Open Internet Explorer and navigate to the following:

http://<dhcp_ip_address>:8080/download

where <dhcp_ip_address> represents the IP address that was noted in step 1b.

Note: Do not setup Internet Explorer at this time if you are prompted to do so.

3. Click on the link for **SQL_Windows-<version>.msi**, and save it to the Desktop.

Note: In this example, version 2.0.2-1 was used.

4. Close Internet Explorer when done.
5. Double-click on the msi file on the desktop to perform the installation.
6. When the installation completes, delete the msi file and empty the Recycle Bin.
7. Restore any files or folders that were backed up at the beginning of this section and assign any necessary permissions to those files and folders.

Verify that the Paging File was Re-Created Correctly with a New Volume

To verify that the paging file will be re-created with a new volume perform the following test:

1. Close the graphical console.
2. Right-click the application within the CA AppLogic GUI, and click **Stop** in the context menu.
3. Click the application in the CA AppLogic GUI to open the Application Editor.
4. Click the **Application** menu and click the **Manage Volumes** menu item in the context menu.
5. Select the **data** volume and click the **Delete** button. When prompted to delete the volume and its references, click **OK**.

Note: The **data** volume will be recreated to test to make sure the MSI install was successful.

6. Click the **Add** button to add a new volume, and specify the following:
 - Name: **data**
 - Size: **8G**
 - Filesystem: **ntfs08**
7. Click **Next** in the Create Volume dialog. Wait for the volume to be created as it can take awhile.
8. Click the **Close** button in the Create Volume dialog.
9. Click the **Done** button in the Manage Volumes dialog.
10. Right-click the singleton in the Application Editor, and select **Modify Boundary** in the context menu.
11. Click the **Volumes** tab.
12. Select the **SQL_ISO** volume and click the **Delete** button. When prompted to remove parameterization, click **OK**.
13. Click the **OK** button in the Class Definition window.
14. Click the **Save** button in the Application Editor and close the Application Editor.
15. Right-click the singleton in the Application Editor, and click **User Volumes** in the context menu.
16. Click the field under the **App Volume** column for the **data** volume, select **data** from the drop-down list.
17. Click the **OK** button in the Instance Settings window.
18. Click the **Save** button in the toolbar of the Application Editor, and close the Application Editor.
19. Right-click the application in the CA AppLogic GUI, and click **Start** in the context menu.
20. Wait for the application to finish restarting then, with the application selected, click the **Login (graphic)** button within the toolbar of the CA AppLogic GUI to open the graphical console.
21. Login to Windows using the appropriate credentials. If the password needs to be reset perform the following steps:
 - a. With the application selected within the CA AppLogic GUI, click the **Login (ssh)** button within the toolbar.
 - b. Execute the following command and close the SSH shell:

```
net user administrator <new_password>
```
22. Verify there are no messages that popup.

What if I See a Message Saying a Temporary Paging File was Created?

If you see this message it indicates that the paging file exists on the boot drive instead of the D: drive. Perform the following steps to remove the paging file from the boot drive, and create a paging file on the D drive:

1. Click the **OK** button in the System Properties dialog stating that a temporary paging file was created.
2. Click the **Change...** button in the **Virtual Memory** section.
3. With the **C** drive selected, verify that **No paging file** is selected, and click the **Set** button. When prompted to continue, click **Yes**.
4. Select the **D** drive, verify that **System managed size** is selected, and click **Set**.
5. Click the **OK** button in the Virtual Memory dialog.
6. Click the **OK** button in the Performance Options dialog.
7. Close the graphical console.
8. Right-click the application within the CA AppLogic GUI, and click **Restart** in the context menu.
9. With the application selected, click the **Login (graphic)** button within the toolbar of the CA AppLogic GUI to open the graphical console.
10. Login to Windows with the appropriate credentials.
11. Click the **Server Manager** button in the taskbar of the Desktop.
12. Expand **Storage**, and click **Disk Management**.
13. Verify under the **Status** column of the **data** volume, you see **Page File** as one of the options. Close the Server Manager application when done.

Step 6: Script the Operations to License MS SQL Server Upon Startup

This step is **optional**.

If you setup Microsoft SQL Server to be licensed by end users, you may want to consider developing scripts to automate the process later. Although this can be done manually, by writing a script, you can **provide a standard way for end users to license Microsoft SQL Server**. For this purposes of this document we are using the field developed Microsoft License Key Injection Utility which utilizes a new boundary property – `sql2008r2_prod_key` - to inject the appropriate licensing details for the appliance. This script for this utility is stored in the `C:\appliance\init\mspk` folder and it generates log messages to a file stored in the `C:\appliance\init\logs` folder. It also uses a common utilities script that is stored in the `C:\appliance\init\util` directory. For more information about the Microsoft License Key Injection Utility as well as commands that can be called to license Microsoft SQL Server see [Appendix A](#).

If you plan on using scripts to license Microsoft SQL Server, create and implement those scripts now.

Step 7: Prepare the MS SQL Server Singleton to be Added to a Catalog

As we did previously for the WIN0864S and Windows Filer appliances, we now need to clean up the new MS SQL appliance singleton to prepare it to become a CA AppLogic appliance. This includes the following steps:

1. Delete the file MS SQL Server Configuration File.

This file, if it still exists, can be found at:

D:\ConfigurationFile_x64_R2.ini

2. Enable the necessary protocols for Microsoft SQL Server by doing the following:
 - a. Open the SQL Server Configuration Manager by executing **Start > All Programs > Microsoft SQL Server 2008 R2 > Configuration Tools > SQL Server Configuration Manager**
 - b. Expand the SQL Server Network Configuration node and click **Protocols for MSSQLSERVER**
 - c. Right-click any of the protocols that need to be enabled, and click Enable in the context menu.
Note: Consider what protocols need to be enabled, but make sure that at least TCP/IP is enabled.
 - d. Start or Restart SQL Server by clicking the SQL Server Services node, right-clicking SQL Server (MSSQLSERVER), and selecting **Start** or **Restart** in the context menu.
 - e. Close the SQL Server Configuration Manager window.
3. Since we configured Windows in this example to be licensed and activated by end users, open a Command Prompt and execute the following command; otherwise, skip this step:

```
s1mgr.vbs /upk
```

Note: Click the **OK** button when the message **Uninstalled product key successfully** is displayed.

4. Restart Windows.
Note: The graphical console will become unusable, so you will need to login again by selecting the application within the CA AppLogic GUI, and clicking the **Login (graphic)** button in the tool bar to open a new graphical console. Login to Windows as usual
5. Perform the steps listed in [Step 8: Prepare the Singleton to be Added to a Catalog](#) earlier in this document.
6. Close the graphical console if it is open.
7. With the application selected in the CA AppLogic GUI, right-click the application, and click **Stop** in the context menu.
8. Edit the class by performing the following steps:

- Click on the application within the CA AppLogic GUI to open the Application Editor.
- Right-click the singleton and click **Modify Boundary** in the context menu.
- Modify the fields on the tabs as follows:

On the **General** tab, do the following:

- Change the Name is to **SQL0864S**.
- Change the Category to **Database Appliances**.
- Change the Description. For example:
 SQL Server Appliance – based on Windows Server 2008 Standard Edition R2 64-bit and SQL Server 2008 R2 SP1 64-bit
- Change the color to **Red**.
- Optionally change the Size as desired.

Click the **Interfaces** tab and verify that the **External Interface** is unchecked.

Click the **Properties** tab and add the following optional properties by clicking the **Add** button, clicking the field, and typing or selecting the appropriate value:

Name	Type	Default	Constraints
sql2008r2_prod_key	String	(empty)	
read_only	String	off	Check Allowed values and type the following: on off
sa_password	String	(empty)	
user_db_name	String	(empty)	
user_login	String	(empty)	
user_password	String	(empty)	
max_connections	Int	0	Min=0, Max=999999

Note: For constraints, click the **Constraints...** button under the **Options** column to bring up the Property Constraints dialog.

- Click the **OK** button in the Class Definition window.

- e. When prompted to save the application click **OK**.
9. Right-click the singleton, and click **User Volumes** in the context menu.
10. Click in the field under the **App Volume** column, for the **data** volume, and select the blank option in the drop-down list.
11. Click **OK** in the Instance Settings window.
12. Click the **Save** button in the toolbar of the Application Editor.

Step 8: Add the Singleton to a Catalog

Now you are ready to transform this singleton in a MS SQL Server appliance that can be instantiated on demand. To do this:

1. With the Application Editor opened, select a catalog to store the singleton in (for example, `system_ms`)
2. Drag and drop the singleton to this catalog.

If the Database Appliances category that was specified in the boundary of this singleton does not already exist within the catalog, it will automatically be created and the new appliance will be stored under the Database Appliances category.

Step 9: Verify your MS SQL Server 2008 Appliance

The final step is to verify that the appliance was correctly built and that upon startup of a new application, that you can log into MS SQL Server. Perform the following steps:

1. In the CA AppLogic GUI, click the **New application** button and provide an application name.
2. Open the Application Editor for the new application by clicking on the new application.
3. Click on the **Application** menu in the Application Editor, and click **Manage Volumes** in the context menu.
4. Add a new volume by clicking the **Add** button, and providing the following information, and clicking the **Next** button when done:
 - Name: `data`
 - Size: 8G
 - Filesystem: `ntfs08`
5. After the volume is created, close the **Manage Volumes** dialog by clicking the **Done** button.
6. Drag the **SQL0864S** appliance onto the canvas from the `/system_ms` catalog.
7. Right-click the appliance, and click **User Volumes**.
8. Click in the field under the **App Volume** column, and select **data** from the drop down menu.
9. Click the **Property Values** tab.

10. Click in the field for the **sa_password** property and type a password for the sa user.
11. Click the **OK** button when done.
12. Click the **Save** button in the toolbar of the Application Editor, and close the Application Editor.
13. Right-click the application within the CA AppLogic GUI, and click **Start** in the context menu.
14. Log into the appliance using the graphical console by clicking the **Login (graphic)** button in the toolbar.
15. Open **Microsoft SQL Server Management Studio** by clicking **Start > All Programs > Microsoft SQL Server 2008 R2 > SQL Server Management Studio**
16. Log in as the **sa** user.
17. Verify the system databases exist by expanding **Databases > System Databases**.

Chapter 5: Creating a Microsoft IIS Appliance

In the previous chapter we demonstrated how our sample WIN0864S appliance can be used to create a Microsoft SQL Server 2008 r2 appliance. In this chapter we are going to demonstrate how the same appliance can be used to build a Microsoft IIS appliance.

Note: Although we are basing the new appliance on the WIN0864S Windows 2008 R2 64 bit Standard Edition server, these steps can also be adapted for use with other Windows Server 2008 R2 editions.

Following is a list of steps we will follow in this chapter:

- [Step 1: Create AppLogic Application](#)
- [Step 2: Install IIS](#)
- [Step 3: Install IIS MSI](#)
- [Step 4: Prepare singleton to be added to catalog](#)
- [Step 5: Add singleton to catalog](#)
- [Step 6: Verify IIS appliance](#)

For more information on these procedures refer to the *CA AppLogic Appliance Developer Guide*, section IIS08W/IIS08S/IIS08E/IIS08DC: Installation Reference which is found at the following link:

http://doc.3tera.com/AppLogic30/en/Developer_Guide/DistInstallIIS08W_1.html

Step1: Create the AppLogic Application to Install IIS

The first step is to create a new application which will serve as the workspace for building a new IIS appliance. After creating the application, we will drag and drop an instance of the WIN0864S appliance from the catalog onto the application canvas and prepare it to install IIS.

To create the application, do the following:

1. Login to the CA AppLogic GUI dashboard, and click the **Applications** tab.
2. Click the **New application** button in the tool bar of the CA AppLogic GUI.
3. Provide a name for the new application and click **OK**.
4. Click on the application to open the Application Editor.
5. Select the **system_ms** catalog from the drop-down menu in the upper left-hand corner of the Application Editor to change the catalog.
6. Drag and drop the **WIN0864S** appliance from the catalog onto the canvas.

7. Right-click the appliance and click **Branch Class** from the context menu.
8. When prompted to save the application, click **OK**.
9. Right-click the singleton, and click **Attributes** in the context menu.
10. Change the Name to **IIS0864S**, and click the **OK** button within the Instance Settings window.
11. Click the **Save** button within toolbar of the Application Editor, and close the Application Editor.

Step 2: Install IIS

Next, we install IIS by doing the following:

1. Right-click the application in the CA AppLogic GUI and click **Start** in the context menu.
2. With the application selected, login using the graphical console by clicking the **Login (graphic)** button in the toolbar of CA AppLogic GUI.
3. Login to the Windows appliance with the appropriate credentials.
4. Install the IIS role and role services by performing the following steps:
 - a. Click the **Server Manager** button in the taskbar on the desktop.
 - b. Click the **Roles** node within the **Server Manager** window.
 - c. Click the **Add Roles** link.
 - d. Click the **Next** button on the **Before You Begin** page within the **Add Roles Wizard**.
 - e. Check the **Web Server (IIS)** role, and click **Next**.
 - f. Click **Next** on the **Web Server (IIS)** page of the wizard.
 - g. Verify that only the following services are checked, and click **Next**:

Common HTTP Features:

- Static Content
- Default Document
- Directory Browsing
- HTTP Errors
- HTTP Redirection

Application development

- ASP.NET
- .NET Extensibility
- ASP
- CGI
- ISAPI Extensions

- ISAPI Filters
- Server Side Includes

Health and Diagnostics

- HTTP Logging
- Request Monitor

Security

- Basic Authentication
- Request Filtering

Performance

- Static Content Compression

Management Tool

- IIS Management Console
- IIS Management Scripts and Tools
- IIS 6 Management Compatibility (including IIS 6 Metabase Compatibility , IIS 6 WMI Compatibility and IIS 6 Scripting Tools)

- h. Click the **Install** button, on the **Confirm Installation Selections** page of the wizard.
- i. Click the **Close** button when finished.
- j. Close the **Server Manager** window.

Step 3: Install the IIS MSI

The next step is to install the IIS MSI which enables the singleton to be fully managed by CA AppLogic.

Note: When the MSI installs, it deletes the following folders, along with all of their contents, and removes any permissions that may have been set:

- C:\appliance
- C:\cygwin

Therefore, if you have made any changes to either of these folders or their contents, you should back them up now. For example, for the purposes of the examples used in this document, we modified the following:

- C:\appliance\init (and all of its contents)
 - This folder structure contained all of the scripts that were written for the licensing example. This folder will be backed up.
- C:\Cygwin\etc\sysconfig\applogic_appliance

This file contains modifications that were used for the licensing example. This file will not be backed up and will make the modifications to it again after the IIS MSI re-creates this file.

After making the necessary backups, install the IIS MSI by doing the following:

1. Locate the ip address of the grid controller by performing the following steps within Windows:
 - a. Open a command prompt and execute the following command:

```
ipconfig /all
```
 - b. Review each of the **Ethernet adapter Local Area Connection X** sections within the command prompt, and look for the property **DHCP Server**. The Local Area Connection that contains this property is the default interface. Note the IP address of the DHCP server.

Note: In this example, Local Area Connection 3 contained the DHCP Server property, although it may be different for other implementations.
2. Open Internet Explorer and navigate to the following location:
http://<dhcp_ip_address>:8080/download
Where <dhcp_ip_address> represents the ip address that was noted in step 1b.
Note: Do not setup Internet Explorer now if it prompts you to do so.
3. Click on the link for **IIS_Windows-<version>.msi**, and save it to the Desktop.
Note: In this example, version **2.0.2-1** was used.
4. Double-click on the msi file on the desktop to perform the installation.
5. Delete the msi file and empty the Recycle Bin.
6. Restore any files that were backed up at the beginning of this section and assign any permissions that were needed for those files/folders again.

Step 4: Prepare the Singleton to be Added to a Catalog

The next step is to prepare the IIS singleton by modifying the boundary properties and cleaning up extraneous files and settings before it is copied into the catalog as an appliance. Perform the following steps:

1. Review and complete the standard preparation steps documented earlier for the Windows appliance (see [Step 8: Prepare the Singleton to be Added to a Catalog](#)).
2. When these are complete, close the graphical console if it is still open.
3. Right-click the application in the CA AppLogic GUI and select **Stop** in the context menu.
4. Edit the class by performing the following steps:
 - a. Open the Application Editor, if it is not opened already, by clicking on the application within the CA AppLogic GUI.

- b. Right-click the singleton and select **Modify Boundary** in the context menu.
- c. Make the following changes on the tabs:

On the **General** tab, do the following:

- Change the Name to **IIS0864S**
- Change the Category to **Web Servers**.
- Change the description. For example:
 “Web Server Appliance – based on Windows Server 2008 Standard Edition R2 64-bit, IIS, and ASP.NET”
- Verify the color is set to **Blue**.
- Optionally change the size as desired.
- Verify the OS Icon is set to **Windows**.

Click the **Interfaces** tab, and do the following:

- Verify that the **External Interface** is unchecked.
- Rename the **out** terminal to be called **db** by clicking in the field under the **Name** column to show the text editor.
- Verify that the **net** terminal is marked as **Gateway** under the **Options** column.
- Add the following terminals by clicking the **Add Out Terminal** button and clicking in the fields to change the values:

Terminal Name	Direction	Protocol
Fs	Out	any
log	Out	any
aux	Out	any

- Use the up and down arrows to the right of the interfaces list to reorder the interfaces as follows:
 1. in
 2. db
 3. fs
 4. log
 5. aux

6. net

7. mon

Click the **Volumes** tab, and add a new volume by clicking the **Add** button and specifying the following information:

- Type: Placeholder
- Name: content
- Click the field under the **Mount on** for the **content** volume and type the following:

D:\

Click **Next** when done:

Click the **Properties** tab and add the following properties by clicking the **Add** button and clicking in the fields to type or select a value:

Notes: To set a constraint, click the **Constraints...** button under the **Options** column.

Name	Type	Default	Constraints
content_on_fs	String	off	Check Allowed values and type the following: on off
docs_dir	String	/	Check Filter and type the following: ^[a-zA-Z0-9\s/_-]*
logs_enabled	String	off	Check Allowed values and type the following: on off
logs_base_dir	String	/	Check Filter and type the following: ^[a-zA-Z0-9\s/_-]*
index_files*	String	default.htm default.asp default.aspx index.html	Check Filter and type the following: [a-zA-Z\.\s]*
max_connections	Int	0	Check Min-Max and set the following: Min=0, Max=999999

Name	Type	Default	Constraints
idle_timeout_sec	Int	15	Check Min-Max and set the following: Min=1, max=3600
data_timeout_sec	Int	300	Check Min-Max and set the following: Min=1, max=3600

Notes:

- **content_on_fs** and **docs_dir** properties are vital to properly configuring the Home Directory properties of the web service. Content_on_fs is mandatory and indicates whether the volume containing the site's content will be local or remotely accessed. The docs_dir property indicates the name of the directory on that volume where the site's files are located. Always set these values using AppLogic properties. If you were to set these properties in the Web Site Properties dialogue at run time, they will be reset when the appliance is restarted.
- If the property **content_on_fs** is set to **on**, then IIS will look for the content on a remote server. [\\fs\share](#) is considered the default directory for the content. Whatever is specified in the docs_dir property, will be appended to this default directory. For example, if the docs_dir property is set to **/mywebapp** then the default directory will be set to [\\fs\share\mywebapp](#) . If the docs_dir entry is left empty or set to **/** then the site's home directory will be the root of \fs\share.
- Later when configuring the instance, the path that is specified for both the **docs_dir** and **logs_base_dir** properties must exist prior to the appliance starting up. Typically, if it is not desired to store all of the content and logs on the root directory, one would start the appliance specifying the root directory (**/**), create the directory structure as needed, stop the appliance, reconfigure these properties to point to the new existing path, and starting the appliance again.
- * - For the index_files property, those four files mentioned under the value column are considered all on one line with a space in between each file.
- When a NAS device is used for this directory, \fs\share points to the /mnt/data directory. Note that the docs_dir property is specified with forward slashes although it is displayed in the dialog with back slashes.

Click the **Resources** tab and modify the resources as follows:

Resource	Min	Max	Default
CPU (num)	0.25	4	1
Memory (bytes)	768M	32G	2G
Bandwidth (bits/sec)	1M	2G	100M

Note: The resource values listed in the previous table apply to the Standard Edition of Windows. If you are using a different edition of the Windows operating system consult the following document for resource recommendations:

http://doc.3tera.com/AppLogic30/en/Developer_Guide/index.htm?toc.htm?DistInstallISO8w_1.html

5. Click the **OK** button in the Class Definition window.
6. When prompted to save the application click **OK**.

Step 5: Add the Singleton to a Catalog

You are now ready to transform this singleton into an appliance by performing the following step:

1. Open the application editor
2. Drag and drop the singleton to a Catalog (for example, **system_ms**).

Note: You must assign yourself full access rights to the catalog before you can move the singleton. See [catalog modify_acl \(Catalog Management\)--Replace Portion of Global Catalog ACL](#) in the Command Line Shell Reference Guide for more information

The ISO864S appliance is now ready to be used.

Step 6: Verify the IIS Appliance

Now that the appliance has been built, we can test it by verifying that a new application that contains the IIS appliance can start. Another test is to create a simple web page and place it on the content volume to verify that IIS will serve the web page. To perform these tests, do the following:

Note: The following test only verifies the IIS features – it does not test the licensing features as that would require either a NET gateway or the external interface to be enabled.

1. In the CA AppLogic GUI, click the **New application** button and provide an application name.
2. Click on the application's name to open it in the Application Editor

3. Click on the **Application** menu in the Application Editor, and select **Manage Volumes** in the context menu.
4. Add a new volume by clicking the **Add** button, and providing the following information:
 - Name: content
 - Size: 1G
 - Filesystem: ntfs08Click **Next** button when you are done.
5. After the volume is created, close the **Manage Volumes** dialog by clicking the **Done** button.
6. Drag and drop the **IIS0864S** appliance from the **/system_ms** catalog to the canvas.
7. Right-click the appliance, and click **User Volumes**.
8. Click in the field under the **App Volume** column, and select **content** from the drop down menu.
9. Click the **Property Values** tab.
10. Verify that the value for **content_on_fs** is set to **off** and that value for **docs_dir** is set to **/**.
11. Click the **OK**
12. Click the **Save** button in the toolbar of the Application Editor, and close the Application Editor.
13. Right-click the application within the CA AppLogic GUI, and click **Start** in the context menu.
14. Log into the appliance using the graphical console by clicking the **Login (graphic)** button in the toolbar.
15. Login to the Windows appliance using the appropriate credentials.

To test the IIS configuration first create a simple **default.htm** file (with some text in it) in the docs_dir directory (/ by default, which is D:\). Then, launch **http://localhost** on the IIS server. This should display the default.htm page.

Note: By default, Windows hides the extension of files of known types. If you create a new text file, and rename it to “default.htm”, the actual file name will be called **default.htm.txt**, which will not show up when navigating to <http://localhost>.

Chapter 6: Application Overview and Design Guidelines

Once they have been added to the catalog CA AppLogic appliances can be dragged and dropped onto the Application canvas to create full functioning applications that can be deployed on the grid. In chapters 7 and 8 we will demonstrate how to build two new applications based on the sample appliances we have created thus far, however, let us first review some basic design guidelines and considerations. Note that, many of the guidelines provided for appliances in Chapter 2 apply to applications as well.

For more information on creating AppLogic applications, see the following:

http://doc.3tera.com/AppLogic30/en/User_Guide/1477618.html

Application Design Considerations

In addition to understanding the product architecture for the application you are building – including software dependencies and communications requirements – there are additional design issues to consider, such as:

- Where will boundary properties be configured – at the appliance level or at the application level?
- Will you use appliances or singletons?
- Where will the user volumes be stored – with the application or with the appliance?
- Must the appliances in the application start in a specific order?
- How will the application communicate with the outside network?

Application Properties vs. Appliance Properties

Applications consist of appliances and, as noted earlier in this document, both applications and appliances are configured through their boundaries. Although the steps to define boundary properties are similar in both cases, it is important to understand how where they are defined impacts the function of the application and what you can do with it. In short:

- When property values are defined at the *appliance level*, they become transparent to the appliance's boot environment and can be leveraged in a script. Property values can be exported as environment variables in the scripts and these values can also be passed to other scripts that can perform any number of configuration tasks.

- When property values are defined at the *application level* they can be inherited by individual appliances, which can then make this property value available to the appliance level environment. Consequently, applications which contain properly built appliances (this includes the scripts required to configure the application) essentially configure themselves by utilizing the inheritance model to configure their underlying appliances.

Not all properties should be defined at the application level. In general, if properties are considered mandatory for an appliance, or if the value of the property can change with each instance of the application, then these properties should be created at the application boundary with the appliances should be setup to inherit the values. For example, CA Service Desk Manager allows you to configure a public host name through which end users can connect to CA Service Desk Manager when it's behind a firewall. Since the public hostname itself may be different for each instance of an application the public hostname *property* should be defined at the application level.

On the other hand, if a property value is meant to be the same for each instance of an application then you may not want to expose it at the application level. To continue with our previous example, the default port on which CA Service Desk Manager runs is "8080". Since you are unlikely to change this, it does not make sense to expose the port number as a configurable application property.

There are also security reasons for not exposing properties at the application level. For example, as part of the SQL0864S appliance we created earlier we defined a property that is used to configure the sa user's password. If this property was defined at the application level, then anyone who has access to configure the application will be able to view this password. It takes greater permissions to open the application and to view the configuration of an appliance.

In the application creation example chapters to follow we are going maximize the agility value of the applications by setting values for all mandatory properties at the application level.

Singletons, Appliances and Data Persistence

Another crucial consideration in the development of applications is the need for data persistence and how that need measures up against the desire to maximize the re-usability of the application. This will determine whether the applications uses appliance classes, singletons or a combination of the two. In general:

- Since *appliances* are stored as templates in either the global or Application (Local) catalogs, they are easy to reuse. All a user has to do to create a new, ready-to-use instance of the appliance is to create a new application, locate the catalog the appliance is stored in, and drag and drop it onto the canvas. However, if changes are made to any of the appliance volumes (like the boot volume) then those changes will be lost if the application is rebuilt or if the grid is upgraded.

- *Singletons*, on the other hand, are only stored in applications – not in a catalog. This means that they cannot be reused in another application. However, any changes that are made to the singleton’s volumes will persist - even if the grid is upgraded or if the application is rebuilt.

Application User Volumes vs. Appliance Volumes

Like boundary properties, volumes can be stored either with the appliance (“Appliance Volumes”), or with the application. As with singletons and appliances, the decision on whether to create an Application User Volume or an Appliance volume mostly comes down to the need for data persistence and reusability. In short:

- Application User Volumes provides better data persistence, but are more difficult to re-use.
- Appliance Volumes provide better reusability, but make it harder to persist the data.

In the previous chapters, we created appliances that used volumes of type “placeholder”. Volumes were then created at the application level and later used to configure the appliance. Because the volume is stored with the application, data persistence will be enabled on these volumes only, even if the grid is upgraded or if the application is rebuilt. However, if a new application is created, then new application user volumes will also have to be created for the new application.

If we had, instead, chosen to create those volumes and store them with the appliance in the catalog, when those appliances were dragged and dropped from the catalog onto the canvas to create a new application the appliance volumes would have already been created. However, if the application is rebuilt, or if the grid is upgraded, all changes to those appliance level volumes is lost.

There are workarounds to each approach. For example, if you want to make an Application User Volume reusable, you can copy it from one application to another. As well, if you want to enable data persistence on appliance volumes you could, after dragging an instance of the appliance onto the canvas within the application, branch that appliance to create a singleton.

The Importance of Appliance Start Order

Each instance of an appliance within an application has a field called “Start Order” that defines when the instance should be started upon startup of the application, in relation to the other appliances in the application. The lower the number that is specified for this field, the sooner the instance will start.

For example, consider the start order configured for the following appliances:

Appliance	Start Order Value
IN	10
IIS0864S	30
SQL0864S	20
NET	10

Upon startup of the application, both the IN and the NET gateway instances will be ordered to start first at virtually the same time since they have both the lowest start order and they have the same order (“10”). The SQL0864S instance (“20”) will be started next, followed by the IIS0864S instance (“30”) which will be started last.

Upon shutdown of the application, the start order is also used to determine what order the instances should be shutdown. Instances will shut down in reverse order of how they started up. So using the same example above, IIS0864S would shutdown first, SQL0864S would shutdown second, and the IN and NET gateway instances will shut down last.

When determining the Start Order of each instance you will need to consider if one instance depends on another instance to be available before it starts up. If it does not depend on another instance, then it can be started first; however, if it does depend on another instance, then it will need to be configured to start after. To demonstrate, consider the following reasons we used to determine the start order for our example:

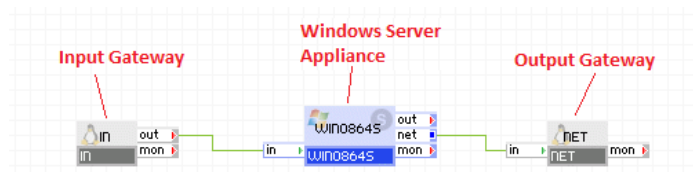
- The IN and the NET gateway instances do not depend on any other instance; therefore, both of these appliances can be started first, at the same time.
- The IIS0864S appliance executes a script upon startup to license Microsoft Windows Server. This requires network access, which means that the NET gateway instance must be started before the IIS0864S instance.
- The SQL0864S appliance also executes scripts upon startup to license both Microsoft Windows Server and Microsoft SQL Server. This requires network access, which means that the NET gateway instance must be started before the SQL0864S instance.
- The IIS0864S appliance also needs to access the user database, which is part of the SQL0864S instance, upon startup. Therefore, the SQL0864S instance must be started before the IIS0864S instance.

Communications between the Application and the Network

When you are designing an application in CA AppLogic it is important to understand how communications are handled both between the individual appliances *within* the application and between the application itself and *other applications* on the network.

Communication between appliances *within* the application is configured through the individual appliance **terminals**, while communications between the application and *other applications* on the network is managed through the use of **gateway appliances**, such as the IN input gateway and NET output gateway.

Consider the following sample CA AppLogic application:



In this example:

- The IN gateway is configured with a public IP address that allows it to receive traffic from the network on certain ports.
- A “virtual wire” connecting the out terminal of the IN gateway with the in terminal on the Windows Server appliances allows traffic to pass from the IN gateway to the Windows Server appliance.
- A second virtual wire connecting the net terminal on the Windows Server appliance to the in terminal on the NET appliance allows traffic to pass from the Windows Server appliance to the NET gateway.
- The NET gateway is configured with a public IP address allowing it to send traffic to the network.

There is no implicit communication into or out of the application nor is there implicit communication between individual appliances within the application. All communication accepted by the application, routed within the application, and forwarded back out of the application must be explicitly defined through the gateways and terminals. In effect, the IN gateway acts as a “firewall” that you must configure to allow traffic. If you do not configure it, no traffic will be allowed in.

In addition, while the Windows Server appliance can initiate a connection to a host on the network by forwarding the traffic through the NET gateway, hosts on the network cannot initiate connections to the Windows Server appliance through the NET gateway, because the NET gateway is connected to the output of the Windows Server appliance, not to its input. Hosts seeking to connect into the Windows Server appliance can only do so through the IN gateway.

Note 1: These rules only apply to communications between terminals of the devices within the boundary interior. These connections are “stateful” - response to a request will be returned through the requesting terminal.

Note 2: By default the IN gateway accepts TCP requests on port 80. So the IN gateway still needs to be configured with an ip address, netmask, and gateway, but if you don’t change the ports or protocols, it will accept only tcp requests on port 80.

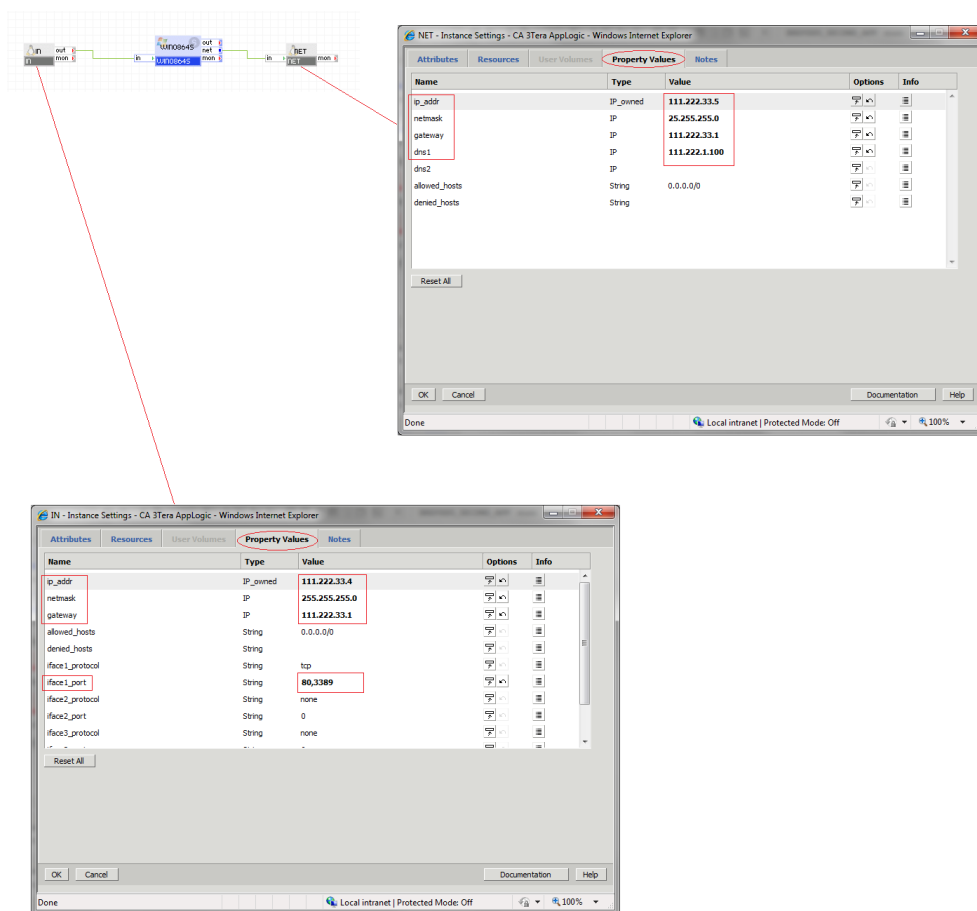
CA AppLogic does allow you to add an external interface to an appliance that will allow it to communicate directly with the outside network, however, this can make the applications less secure and more difficult to maintain than they would be if all traffic were funneled through the input and output gateways. Therefore, use of external interfaces between appliances and the outside network should be limited to those cases where limitations of the product design prevent the application from communicating through the input and output gateways.

Networking setup

Since both the IN (input) and NET (output) gateways have IP addresses on the public network this means that, in order to set up a CA AppLogic application, you will usually need to provide at least two IP addresses from the range of public addresses available to your AppLogic grid:

- one address for the application to receive requests from the external network via the IN gateway, and
- one address for the application to send traffic to the external network via the NET gateway.

The IP addresses for the input and output gateways are configured as part of the appliance’s boundary through their respective Property Values tabs. To illustrate, consider the example on the following page. In this example we have assigned a public IP address (“111.222.33.4”) to the IN gateway through its ip_addr property and we have used the ip_addr property of the NET gateway to assign a different public address (“111.222.33.5”) to that gateway. These are the *only* IP addresses for our application that will be visible on the “public” network (i.e. from outside the application). Requests going *to* the application *from* outside must be sent to address 111.222.33.4, since that is the address we have assigned for the IN gateway to receive traffic. Requests originating *from* the application and sent *to* the public network will appear to outside hosts to be originating from address 111.222.33.5 (i.e., the IP address assigned to the NET appliance).



Design Tip! A limited number of public IP addresses are available on any given AppLogic grid; however, it can be difficult to determine which addresses are in use without examining all of the running applications on the grid and inspecting the properties of each of their gateway appliances. Since an application will fail to start if it attempts to use a public address that is already in use by another application instance or that is not allocated to the grid, it is good practice to use the application’s Description field to note which public IP addresses you are using. Since the Description field is readily visible to anyone browsing the list of applications on the grid, listing your public IP addresses in that field makes it readily apparent which addresses are in use. You can modify an application’s description by right-clicking it in the grid’s Applications tab and selecting Configure, or by selecting Configure from the Application pull-down menu in the application editor. If you are at all unsure about what addresses to use for your applications, consult the administrator of your grid before setting up any public IP addresses.

To complete the network setup for our sample application, we need to know a few more things:

- The network mask (routing prefix) for both the input and output gateways

- The gateway address for both the input and output gateways
- The address of the DNS server, for the output gateway only
- Which port(s) and the respective protocols for those ports to allow requests on, for the input gateway only

The first three items can easily be obtained from your grid's Dashboard tab. For example:

Application IP Range	Netmask	Gateway	DNS Servers
111.222.33.1 - 111.222.33.10	255.255.255.0	111.222.33.1	111.222.1.100
111.222.33.11 - 111.222.33.20	255.255.255.0	111.222.33.1	
111.222.33.21 - 111.222.33.30	255.255.255.0	111.222.33.1	
111.222.33.31 - 111.222.33.40	255.255.255.0	111.222.33.1	

VLAN ID Range
0 - 0

In this example you can see the appropriate **Netmask** and **Gateway** values associated with various ranges of public IP addresses on the grid, and a **DNS Server** address that is common to the whole grid. These values are specified as follows on the Property Values tab for their respective appliance:

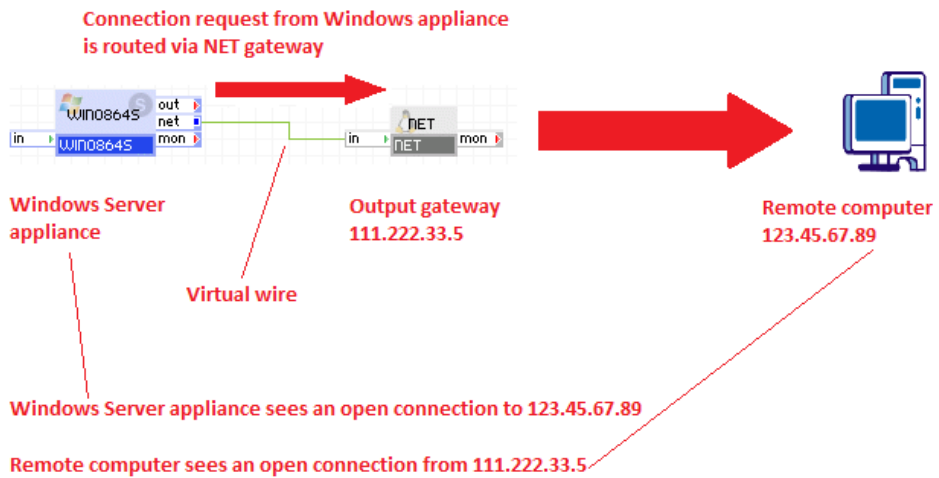
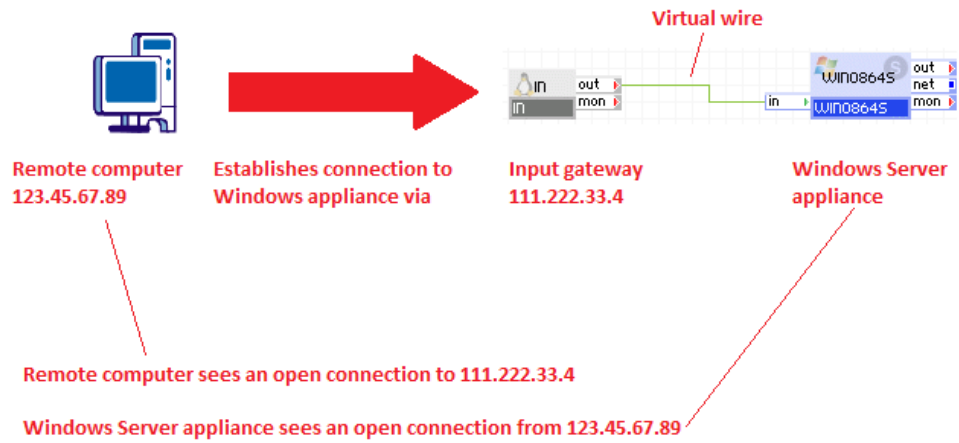
- the **Netmask** value is specified for the **netmask** property of the IN and NET gateways.
- The **Gateway** value is specified for the **gateway** property of both the IN and NET gateways.
- The **DNS Servers** address is specified for the **dns1** property of the NET gateway; the IN gateway does not have this property.

The **iface1_port** property of the IN gateway is where we specify the port(s) that our application will use to receive traffic from outside the application. The settings for this property are application-specific. In our sample application, we've configured our input gateway to allow traffic on port 80, which will allow the application to receive HTTP requests, and on port 3389, which is the port that Windows uses for the Remote Desktop Connection when that feature is enabled. These allowable port numbers are specified in a comma-separated list: **80,3389**. It is important that our application allows traffic from the outside network on port 3389, as this port enables us to open a Remote Desktop Connection to our Windows Server appliance from a physical machine located outside the application. More complex applications may demand the opening of additional ports. For example, you may need additional ports for HTTP requests, database connections, or product-specific communication functions. The **iface1_protocol** property specifies the protocol to be filtered by **iface1_port** property; in our case it is **tcp**.

In terms of how other application interact with the application you are designing consider the following:

- Requests made *to* the application from other hosts on the network must be directed to the address of the application's input gateway, and the input gateway must be configured to allow such requests.
- Requests made *from* the application to other hosts on the network will appear to those hosts to originate from the address of the application's output gateway.

To illustrate:



Chapter 7: Creating a Microsoft IIS Application

This chapter demonstrates how to use the sample appliances we created earlier to build a simple Microsoft IIS web application.

We will be using the following appliances:

- IN – input gateway
- IIS0864S – web server with active content (e.g., CGI scripts)
- SQL0864S– MSSQL database appliance
- NET – External Network Gateway
- MON – Monitoring Appliance

The steps that we will use are as follows:

- [Step 1: Build a new AppLogic Application](#)
- [Step 2: Add appliances to the application](#)
- [Step 3: Connect the appliances within the application](#)
- [Step 4: Add Properties to the Application Boundary](#)
- [Step 5: Creation Application User Volumes](#)
- [Step 6: Configure instance settings for each appliance](#)
- [Step 7: Configure the application](#)
- [Step 8: Test the application](#)

Step 1: Build a new CA AppLogic Application

We'll start by creating a new CA AppLogic application. To do this:

1. Launch the CA AppLogic Graphical User Interface, and login.
2. Click the Applications tab.
3. Click the **New application** button on the toolbar, and provide a name.
4. Click **OK**

Step 2: Add Appliances to the New Application

Next, we add the appliances required to make this a fully functional application. To do this:

1. Click the new application to open the Application Editor.
2. Locate the **Gateways** category under the **system** catalog and drag the **IN** and **NET** appliances onto the Application Editor canvas.
3. Locate the **Monitoring** category under the **system** catalog, and drag the **MON** appliance onto the canvas.
4. Locate the **IIS0864S** appliance we created previously (see [Creating a Base IIS Appliance](#)) in the **system_ms** catalog, and drag it onto the canvas.
5. Locate the **SQL0864S** appliance we created previously (see [Create a Base SQL Server Appliance](#)) under the **system_ms** catalog, and drag it onto the canvas.
6. Save the application, by clicking the **Application** menu in the Application Editor, and clicking **Save** in the context menu.

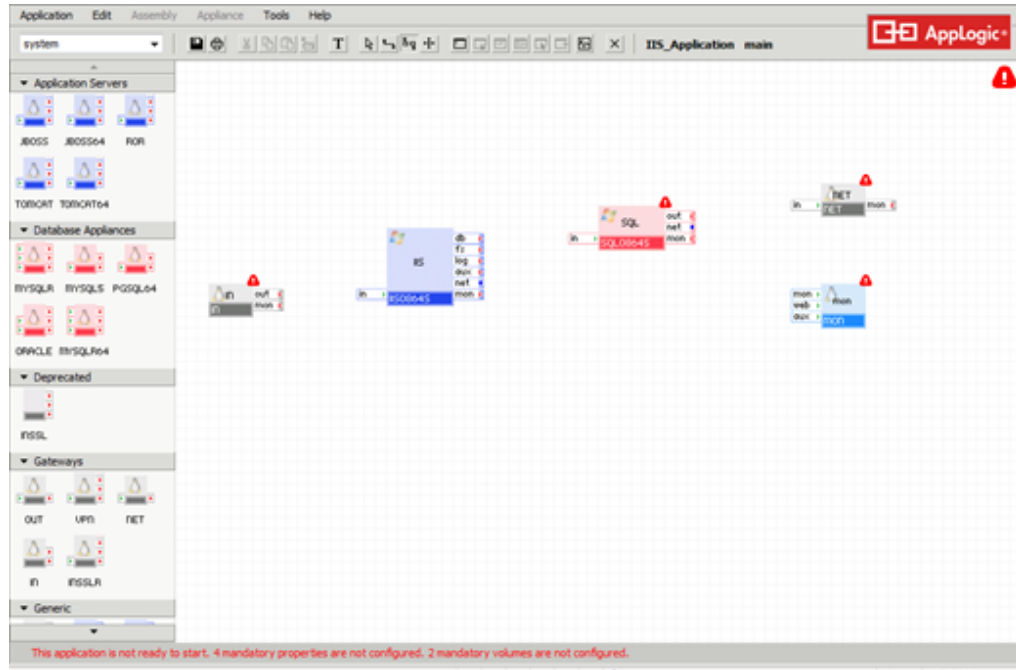
Important! Changes to the application can **only** be saved by selecting **Application > Save** from the menu bar or by clicking the **Save** button on the tool bar. When an asterisk (*) appears next to the name in the Window's title bar this indicates that the application has not been saved.

7. Right-click the **IIS0864S** appliance and select **Attributes** in the context menu.
8. Change the value of the Name field to **IIS**, and click **OK** in the Instance Settings window.
9. Right-click the **SQL0864S** appliance and select **Attributes** in the context menu.
10. Change the value of the Name field to **SQL**, and click **OK** in the Instance Settings window.
11. Save the application by clicking the **Save** button in the toolbar of the Application Editor.

Since CA AppLogic appliances are protected resources within an AppLogic application and are not normally visible to the network domains, it is not necessary to apply standard customer naming conventions to the Windows hostname. Also, since the Windows hostname is set to the appliance instance name by default, the appliance instance name does not necessarily need to follow the standard customer naming conventions either.

If the automatic generation of hostnames feature was disabled (see [Disabling Auto Host Name Generation](#) earlier) you will need to manually change the host names.

The end result should look similar to the following:



Step 3: Connect the Appliances

Next, we need to define how the appliances communicate *within* the application (through terminals) and how communication will be managed *into and out of* the application (through the gateway appliances).

In this example, client requests *to* the application come through the IN gateway which then directs that traffic to the IIS0864S server (via the in terminal) where they are served. Communications *from* either the IIS0864S or SQL0864S appliances to the external network go through the NET gateway by way of their respective net terminals. Within the application the IIS0864S instance connects to the SQL appliance through the db terminal whenever an application on that instance (for example, ASP .NET or ASP) needs to access persistent data.

To connect the appliance instances do the following:

1. Click the **Connect Mode** button on the Application Editor toolbar.
2. Connect the following appliances by clicking on the output terminal of the source appliance to the input terminal of the destination appliance:

Source	Terminal	Destination	Terminal
IN	out	IIS0864S	in
IIS0864S	db	SQL0864S	in

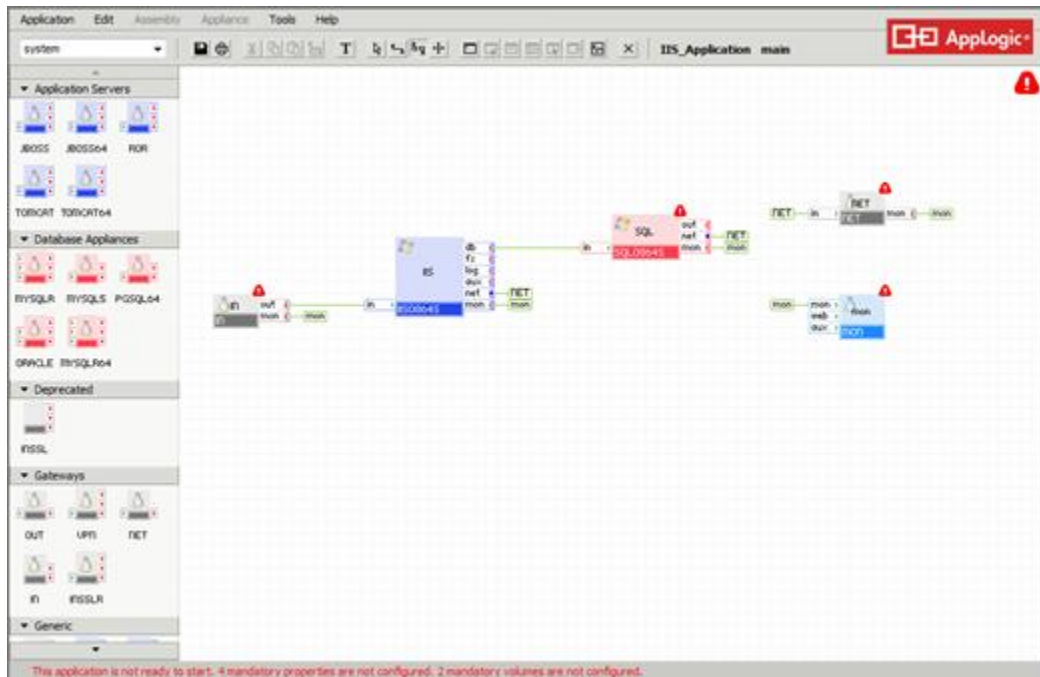
3. Select the **Balloon Mode** button on the Application Editor toolbar.

- Connect the following appliances by clicking on the output terminal of the source appliance(the one making the request) to an input terminal of the destination appliance (the one receiving the request):

Source	Terminal	Destination	Terminal
IN	mon	MON	mon
IIS0864S	net	NET	in
IIS0864S	mon	MON	mon
SQL0864S	net	NET	in
SQL0864S	mon	MON	mon
NET	mon	MON	mon

- Click the **Save** button in the toolbar of the Application Editor.

The result should look like the following:



Step 4: Add Application Properties

Next, we will define application boundary properties so that variable appliance property values can be configured conveniently at the application boundary. To do this, perform the following steps:

1. With the Application Editor open, click the **Application** menu, and select **Modify Boundary** from the context menu.
2. Click the **Properties** tab.
3. Add the following properties by clicking the **Add** button, clicking in the fields and typing the following information.

Note: To make a property mandatory, click the **Mandatory** button under the **Options** column.

Name	Mandatory?
win2008r2_prod_key	No
kms_server_port	No
sql2008r2_prod_key	No
content_on_fs	No
docs_dir	No
IN_ip_addr	Yes
NET_ip_addr	Yes
NET_dns1	Yes
NET_dns2	No
netmask	Yes
gateway	Yes

4. Click **OK** when you are done adding the properties.
5. Click the **Save** button in the toolbar of the Application Editor.

Configuring these properties at the application boundary level enables us to configure the appliances without having to navigate to each appliance boundary. Note that, while the `sa_password` property on the SQL Server appliance could have been made an application boundary property, we decided against for security reasons. For example, if a user configures the application and specifies the `sa_password`, and does not later remove it, anyone who can configure this application will see the `sa_password`. Instead, this value will be configured later when the SQL Server instance is configured.

Note: The `content_on_fs` and `docs_dir` properties were left as optional because if you mark properties as both mandatory and having a constraint, the application will fail to start.

Step 5: Create Application User Volumes

Next, we are going to create user volumes for the application. Due to the way in which both the IIS0864S and SQL0864S appliances were configured, it will not be necessary to branch them as all of the content will reside on application user volumes.

To create the user volumes on the application for the IIS0864S, SQL0864S, and MON appliances do the following:

1. Within the Application Editor, click the **Application** menu, and click **Manage Volumes** in the context menu.
2. Add the following volumes by clicking the **Add** button, providing the information below, and clicking the **Next** button:

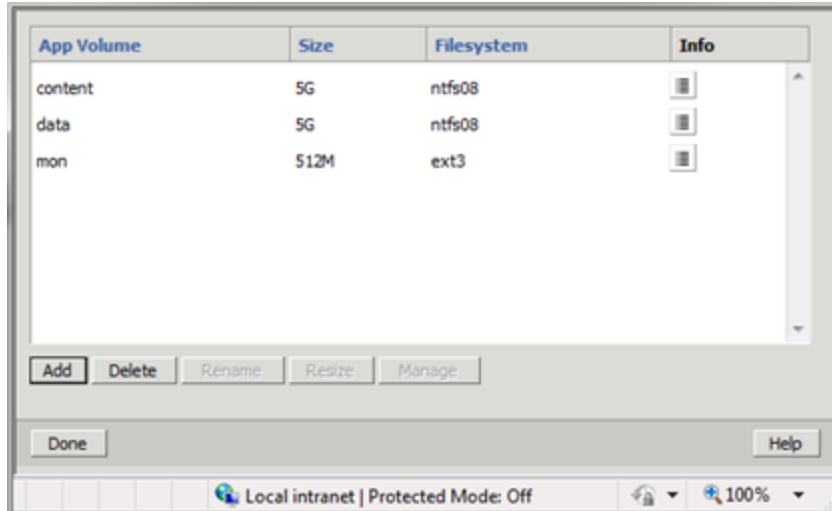
Name	Size	Filesystem
mon	512M*	ext3
data	5G*	ntfs08
content	5G*	ntfs08

* - this values should be appropriate for IIS web application requirements

In this example we chose to use the “ext3” file system since the MON appliance is Linux based.

Note that this step may take awhile to complete.

3. Once finished, verify that the volumes have been created, and click **Done** within the Manage Volumes dialog.



Step 6: Configure the Instance Settings of Each Appliance

We are now ready to configure the instance settings of each appliance.

IN Gateway Appliance

The IN Gateway Appliance must be configured to do the following:

- inherit the property values from the Application properties.
- start first along with the NET Gateway and MON appliances.

To do this:

1. Within the Application Editor, right-click the **IN** appliance, and click **Attributes** from the context menu.
2. Type **10** in the **Start Order** field to signify this appliance will be started up within the first group of appliances.
3. Click the **Property Values** tab.
4. Click the **Redirect to assembly** button under the **Options** column for the **ip_addr** property.
5. Click in the Field under the **Value** column, and select **IN_ip_addr** from the drop-down menu.
6. Repeat steps 4 and 5 for the following properties:

Name	Value
netmask	netmask
gateway	gateway

Notes:

- The **iface1_protocol** and **iface1_port** properties specify the protocol and ports that the appliance will accept requests on. By default, those values are **tcp** and **80**, respectively. It is not necessary to change these values.
- To enable Remote Desktop into the IIS0864S appliance, modify the **iface_1_port** to read:

80, 3389

7. Click **OK** in the Instance Settings window.
8. Click the **Save** button in the toolbar of the Application Editor.

IIS0864S Appliance

The IIS0864S appliance will be configured to do the following:

- Use the Application User Volume called “content”. This user volume will contain the IIS web applications.
- Start last.

The *content_on_fs* property is used to indicate where the content is located - relative to the file system at the fs terminal (value is set to “on”) or on the content volume (value is set to “off”). In this exercise, the *content_on_fs* property will be set to off.

The path specified within the *docs_dir* property must already exist. If the path does not exist, the application will not start up. Typically, one will start the application pointing to the root (/) directory, create a new directory structure on the D drive, stop the application, modify the *docs_dir* property again to point to an existing directory, and restart the application.

Note: For the purposes of this exercise, the Microsoft SQL Server Client does not need to be installed within this appliance. If you decide to install the client, this appliance must be branched first before the install, and the boot volume may need to be resized to accommodate the space needed by the client.

To configure the IIS0864S appliance, do the following:

1. Within the Application Editor, right-click the **IIS0864S** appliance, and click **Attributes** from the context menu.
2. Type **30** in the **Start Order** field to signify this appliance will be started up last.
3. Click the **User Volumes** tab.
4. Click the field under the **App Volume** column for the **content** volume, and select **content** from the drop-down list.
5. Click the **Property Values** tab.

6. Click the **Redirect to assembly** button under the **Options** column for the **win2008r2_prod_key** property, and click the field under the **Value** column to select the **win2008r2_prod_key** property from the drop-down list.
7. Repeat step 6 for the following properties:

Name	Value
kms_server_port	kms_server_port
content_on_fs	content_on_fs
docs_dir	docs_dir

8. Click the **OK** button in the Instance Settings window.
9. Click the **Save** button in the toolbar of the Application Editor.

SQL Appliance

Next, we need to configure the SQL appliance to do the following:

- Use the Application User Volume called “data” to store the SQL Server data.
 - Enable the SQL databases to be accessed.
- Note:** Although properties can be configured at the application level as well, for the purposes of this exercise, we are configuring them only at the appliance level.
- Startup *after* the gateway and mon appliances but *before* the IIS0864S appliance.

To do this:

1. Right-click on the **SQL0864S** appliance and click **Attributes** in the context menu.
2. Type **20** in the **Start Order** field.
3. Click the **User Volumes** tab.
4. Click in the field under the **App Volume** column for the **data** volume, and select **data** from the drop-down list.
5. Click the **Property Values** tab.
6. Click the **Redirect to assembly** button under the **Options** column for the **win2008r2_prod_key** property, and click the field under the **Value** column to select the **win2008r2_prod_key** property from the drop-down list.
7. Repeat step 6 for the following properties:

Name	Value
kms_server_port	kms_server_port
sql2008r2_prod_key	sql2008r2_prod_key

8. Update the properties listed below by clicking in the field under the **Value** column and typing in the value mentioned below:

Name	Value	Notes
read_only	off	Database is read-only; value may be modified
sa_password	<new_sa_password>	Default password is Manager. Specify a new password.
user_db_name	web	Default database for user 'website'
user_login	website	Database user account
user_password	<new_db_password>	Database password
max_connections	0	Manage memory settings automatically

9. Click the **OK** button in the Instance Settings window.
10. Click the **Save** button in the toolbar of the Application Editor.

NET Gateway Appliance

Next, the NET gateway appliance will be configured to do the following:

- Inherit the property values from the Application properties.
- Start first, along with the IN Gateway and MON appliances.

To do this, perform the following steps:

1. Within the Application Editor, right-click the **NET** appliance, and click **Attributes** from the context menu.
2. Type **10** in the **Start Order** field to signify this appliance will be started up within the first group of appliances.
3. Click the **Property Values** tab.

4. Click the **Redirect to assembly** button under the **Options** column for the **ip_addr** property.
5. Click in the field under the **Value** column, and select **NET_ip_addr** from the drop-down menu.
6. Repeat steps 4 and 5 for the following properties:

Name	Value
netmask	netmask
gateway	gateway
dns1	NET_dns1
dns2	NET_dns2

7. Click **OK** in the Instance Settings window.
8. Click the **Save** button in the toolbar of the Application Editor.

MON Appliance

Next, the MON appliance will be configured to do the following:

- Use the Application User Volume called “mon”.
- Start first, along with the IN and NET Gateway appliances.

Note: The properties for this appliance do not need to be configured at this time as the default values for this appliance will be sufficient.

To configure the MON appliance, do the following:

1. Within the Application Editor, right-click the **MON** appliance and click **Attributes** from the context menu.
2. Type **10** in the **Start Order** field to signify this appliance will be started up within the first group of appliances.
3. Click the **User Volumes** tab.
4. Click the field under the **App Volume** column for the **data** volume, and select **mon** from the drop-down list.
5. Click **OK** in the Instance Settings window.
6. Click the **Save** in the toolbar of the Application Editor.

Step 7: Configure the Application

Now that the appliances have been configured it is time to configure property values for the application itself. As noted earlier, property values configured at the application level will, in turn, configure the appliances. Perform the following steps:

1. Within the Application Editor, click the **Application** menu, and click **Configure** in the context menu.
2. Click the **Property Values** tab.
3. Provide the following information by clicking in the field under the **Value** column and typing in the value below:

Name	Value
win2008r2_prod_key	Specify the license key for Windows
kms_server_port	<kms_server_name>:<port>
sql2008r2_prod_key	Specify the license key for the MS SQL Server
content_on_fs	off
docs_dir	/
IN_ip_addr	Specify the IP address to assign to the IN Gateway appliance
NET_ip_addr	Specify the IP address to assign to the NET Gateway appliance.
NET_dns1	Specify the IP address of the primary DNS server the NET Gateway appliance will use
NET_dns2	Optionally, specify an IP address for the secondary DNS server the NET Gateway appliance will use
netmask	Specify the network mask that will be used by both the IN and NET Gateway appliances
gateway	Specify the IP address of the gateway that will be used by both the IN and NET Gateway appliances

4. Click **OK** in the Application Configuration window.
5. Click the **Save** button in the toolbar of the Application Editor.
6. Close the Application Editor.

We now have a simple application that can be used to access the SQL database through simple IIS queries. The next section identifies steps for testing the newly created application.

Step 8: Test the Application

Now that the AppLogic application has been created, it is time to verify that it works. To do this we are going to execute a simple test to determine if the IIS node can connect to the SQL database, and then we will verify that IIS can serve a web page. To do this, perform the following steps:

1. Within the CA AppLogic GUI, verify that the Applications tab has been selected.
2. Right-click the IIS application that was created and click **Start** in the context menu.
3. If the Application Editor is not open, click on the IIS application to open the Application Editor.
4. Click the **IIS0864S** node, and click the **Login (graphic)** button in the toolbar of the Application Editor.
5. Log into Windows with the appropriate credentials.

Note: If you need to reset the administrator password, with the IIS0864S node selected, click the **Login (ssh)** button within the toolbar of the Application Editor and execute the following command:

```
net user administrator <new_password>
```

6. Open the **ODBC Data Source** application by executing **Start > Administrative Tools > Data Sources (ODBC)**.
7. Click the **System DSN** tab.
8. Click the **Add...** button.
9. Select **SQL Server** for the driver and click the Finish button.
10. Provide the following information on the first page of the wizard, and click **Next**:

- Name: <Any name is fine>
- Server: **db**

Note: **db** is the name of the out terminal that is on the boundary of the IIS0864S appliance. This is used instead of the name of the server on which SQL Server is installed.

11. Select the option **With SQL Server authentication using a login ID and password entered by the user**, then provide the credentials for the **website** database user, and click Next.

Note: When configuring the instance settings for the IIS0864S appliance, the website user and password were specified on the Property Values tab.

12. Check the **Change the default database to:** checkbox, and select **web** from the drop-down list. Click **Next**.
13. Click the **Finish** button.
14. Click the **Test Data Source...** button and verify the message “**TESTS COMPLETED SUCCESSFULLY!**” is displayed.
15. Click the **OK** button.
16. Click the **OK** button in the **ODBC Microsoft SQL Server Setup** dialog.
17. Verify the System DSN that was just created is selected in the **ODBC Data Source...** dialog and click the **Remove** button to delete it. When prompted to confirm to delete it, click the **Yes** button.
18. Click the **OK** button in the **ODBC Data Source...** dialog to close the ODBC Data Source Application.
19. Test the IIS configuration by creating a simple default.htm(with some text in it) file in the docs_dir directory and launch **http://localhost** on the IIS server which should display the default.htm page. Then, launch the page from outside AppLogic using the URL **http://<IN_ip_addr>** and you should once again see the default web page presented.

Note: By default, Windows hides the extension of files of known types. If you create a new text file, and rename it to default.htm, the actual file name will be called **default.htm.txt**, which will not show up when navigating to <http://localhost>

Chapter 8: Creating a CA Service Desk Application

In the previous chapter we used our sample appliances to demonstrate how to build a simple, application. In this chapter we are going to demonstrate how those same WIN0864S and SQL0864S appliance classes can be used to create a more complex *two*-tiered application on which CA Service Desk Manager r12.6 will be installed. In this example, the CA Service Desk Manager Primary Server will be installed on a WIN0864S appliance, and the MDB will be installed on a SQL0864S appliance.

The steps include:

- [Step 1: Creating the Application](#)
- [Step 2: Branching the classes](#)
- [Step 3: Resizing disks](#)
- [Step 4: Modify Class Boundaries](#)
- [Step 5: Add properties to the Application Boundary](#)
- [Step 6: Add the Application User volumes for the MON appliance](#)
- [Step 7: Modify instance settings](#)
- [Step 8: Configure the Application](#)
- [Step 9: Start the Application](#)
- [Step 10: Login to the Singletons](#)
- [Step 11: Prepare Windows to install CA Service Desk Manager r12.6](#)
- [Step 12: Install the MDB](#)
- [Step 13: Install the Microsoft SQL Server Client](#)
- [Step 14: Install CA Service Desk Manager](#)
- [Step 15: Create scripts for the SQL Client and CA Service Desk Manager](#)
- [Step 16: Prepare the singletons to become appliances](#)
- [Step 17: Move the appliances to the AppLogic catalog](#)
- [Step 18: Verify and finalize the application](#)

Note: Although a product license is not required to install CA Service Desk Manager, once the designated grace period has elapsed you will not be able to use it without providing a license. For the purposes of this example we are not including steps to add the license.

Step 1: Create the Application

As with the previous example, the first step is to create an application with all of the necessary appliances, and connect each of them together so that the appliances can communicate with one another. To do this:

1. Create a new application, and drag and drop the following classes onto the Application Editor canvas.

From the **/system** catalog:

- Gateways Category: **IN**
- Gateways Category: **NET**
- Monitoring Category: **MON**

From the **/system_ms** catalog:

- Operating Systems Category: **WIN0864S**
- Database Appliances Category: **SQL0864S**

2. Click the **Connect Mode** button on the Application Editor toolbar.
3. Connect the following appliances by clicking on the output terminal of the source appliance(the one making the request) to an input terminal of the destination appliance(the one receiving the request):

Source	Terminal	Destination	Terminal
IN	out	WIN0864S	in
WIN0864S	out	SQL0864S	in
WIN0864S	net	NET	in
SQL0864S	net	NET	in

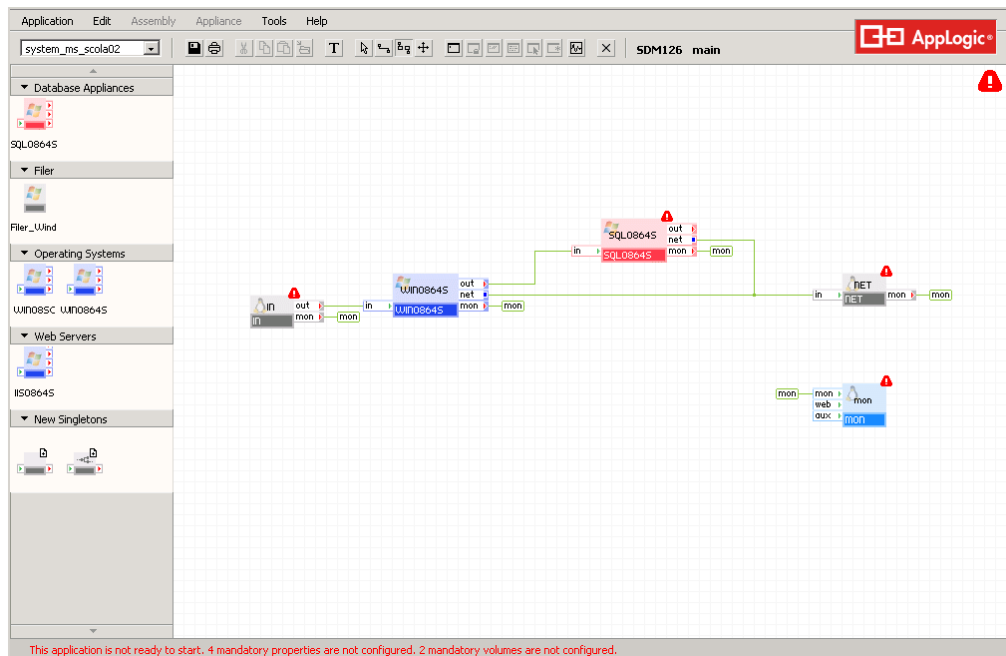
4. Select the **Balloon Mode** button on the Application Editor toolbar.
5. Connect the following appliances by clicking on the output terminal of the source appliance(the one making the request) to an input terminal of the destination appliance (the one receiving the request):

Source	Terminal	Destination	Terminal
IN	mon	MON	mon

Source	Terminal	Destination	Terminal
IIS0864S	mon	MON	mon
SQL0864S	mon	MON	mon
NET	mon	MON	mon

- Click the **Save** button in the toolbar of the Application Editor.

The result should look like the following:



- Click the **Save** button in the toolbar of the Application Editor.

Step 2: Branch the Classes

Next, we are going to create new appliances for the CA Service Desk Manager Primary Server, and for the Database Server that will contain the MDB. We are also going to create a new IN gateway appliance – based on the existing IN – that will be customized for the CA Service Desk Manager r12.6. This requires us to branch the existing appliances.

To do this right-click on each of the following appliances and select **Branch Class** from the context menu:

- IN
- WIN0864S
- SQL0864S

Step 3: Resize the Disks

The CA Service Desk Manager primary server which will be installed on the WIN0864S appliance also requires the installation of the Microsoft SQL Server 2008 R2 client. As a result, we now need to re-size the boot volume on the WIN0864S singleton from 16 GB to 19 GB.

Note: During the development of this document it was found that certain components of the Microsoft SQL Server 2008 client were required to be installed on the boot drive. Additional tests are being conducted to determine if all of the components of Microsoft SQL Server 2008 R2 client can be installed on another drive. If that is true, this step will no longer be necessary - as all of the components of Microsoft SQL Server Client will be able to be installed on another drive and this document will be updated to reflect that change.

You can re-size appliance disk volumes using either the Grid ssh shell or by performing the following steps in the CA AppLogic web interface:

1. Select the **WIN0864S** singleton.
2. Click the **Appliance** menu in the menu bar.
3. Click the **Modify Boundary** menu item.
4. Click the **Volumes** tab in the Class Definition window.
5. Select the **boot** volume and click the **Resize** button.
6. Change the value in the **Size** field to **19G** and click **Next**.
7. Click **OK** in the **Resize Volume** window, and then again in the Class Definition window.
8. Click the **Save** button in the toolbar of the Application Editor.

Note: If you need to resize a disk while the application is running, you will need to restart the application for changes to take effect.

Tip: A pagefile.sys may be created on the boot volumes. Be sure to factor it into your disk size calculation.

Troubleshooting

If you receive the following error message while you are resizing the boot volume...

```
ailed[sic] to copy filesystem - status: 100 -“
```

followed by

```
Failed to resize volume
```

....this is may be caused by corrupted files created during .NET and ASP.net installation.

These files are of type virtual mount, however, because they cannot be stat'ed normally this causes the Windows filer to fail. These files may be safely removed which will enable the volume resize command to succeed and they may be recreated later if needed.

To remove these files, do the following:

1. Click the **Login (ssh)** button in the toolbar of the Application Editor to log into the grid ssh shell
2. Execute the following command:

```
3t vo1 manage application:appliance.boot
```

3. Execute the bash shell command

```
du -sh ./
```

If you get a “permission denied” or “cannot read directory” message for several of the files in the volume these are OK and can be ignored. Others will fail with an error such as “No such file or directory.” These files must be removed. Typically, they will have file names similar to: `./WINDOWS/assembly/GAC_MSIL/IEExecRemote/2.0.0.0_b037811879`. Although they appear to be files, they must be removed with the `rmdir` command. Once these files are cleaned up, the volume resize should execute normally.

Step 4: Modify Class Boundaries

In [Step 2](#) we branched the IN, WIN0864S, and SQL0864S appliances to create singletons. In this step we will be modifying the boundary of each of these singletons to prepare them to become new appliances:

IN Singleton

By default, the IN gateway accepts requests on port 80. For the purposes of our example we need to change that default port in order to accept requests on ports 3389, 8070, 8080, and 9080. These ports are used by the following services:

- 3389 – Used by Remote Desktop to allow users to RDP into the CA Service Desk Manager primary server.
- 8070 – Used by Support Automation
- 8080 – Used by the CA Service Desk Manager primary server
- 9080 – Used by the CA CMDB Visualizer

By creating a new, customized IN gateway appliance that will automatically accept requests on these port we are making it easier for the future creation of additional CA Service Desk Manager applications. Instead of having to configure the standard IN gateway instance each time to accept requests on certain ports, we can just drag and drop this customized IN appliance onto the canvas.

To modify the IN singleton, do the following:

1. Right-click the **IN** singleton, and click **Modify Boundary** in the context menu.
2. Click the **General** tab and make the following changes:
 - Change the *Name* to **SDM_IN_12_6**
 - Change the *Instance Name Template* to **SDM_IN**
 - Change the *Category* to **CA_Service_Desk_Manager_12_6 Input Gateway Appliance specialized for CA Service Desk Manager r12.6 - iptables based incoming connection gateway with firewall (v1.0.0-1)**
 - Change the *Color* to **Green**.
 - Change the *Size* to **Medium**
 - Verify the *OS Icon* is set to **Linux**.
3. Click the **Interfaces** tab and verify that the *External Interface* checkbox is checked.
4. Click the **Properties** tab, then click the field under the **Default** column for the **iface1_port** and change the value to:
3389,8070,8080,9080
5. Click the **Resources** tab, and modify the resource values as follows:

Resource	Min	Max	Default
CPU (num)	0.05	4	0.05
Memory (bytes)	96M	2G	96M
Bandwidth (bits/sec)	500M	1G	500M

6. Click the **OK** button in the Class Definition window.
7. When prompted to save the application, click the **OK** button.

WIN0864S Singleton

The WIN0864S singleton will be modified to support the CA Service Desk Manager r12.6 primary server. Since we want this appliance to be reusable (as it will be eventually stored in a catalog) we are going to create a separate appliance volume for the singleton which will allow us to store this appliance in a catalog. This appliance volume will contain the application data. Since we also want the data on the appliance to be persistent –as users are likely to make changes to the volumes after using it - users will be expected to branch this appliance once they drag the appliance onto their canvas of their new application.

We are also going to add a property to the appliance boundary that will be used to configure the external hostname end users will use to connect to CA Service Desk Manager r12.6.

To do this, perform the following steps:

1. Right-click the **WIN0864S** singleton, and click **Modify Boundary** in the context menu.
2. Click the **General** tab and make the following changes:
 - Change the *Name* to **SDM_PRI_12_6_S**.
 - Change the *Instance Name Template* to **PRI**.
 - Change the *Category* to **CA_Service_Desk_Manager_12_6**.
 - Change the *Description* to: **CA Service Desk Manager r12.6 Primary Server Appliance - based on Windows Server 2008 Standard Edition R2 SP1 64-bit; not activated; CA build (v1.0.0-1)**
 - Change the *Color* to **Green**.
 - Change the *Size* to **Large**.
 - Verify that the *OS Icon* is set to **Windows**.
3. Click the **Interfaces** tab, then click the field under the **Name** column for the **out** interface, and change it to **mdb** to rename the terminal.
4. Click the **Volumes** tab and add the following volumes by doing the following:
 - a. Click the **Add** button
 - b. Specify the information below, and click **Next**:

Type	Name	Size	Filesystem
Instantiable	apps	15G	ntfs08
Placeholder	SQL_ISO	N/A	N/A
Placeholder	SDM_ISO	N/A	N/A

- c. When finished, click on the fields under the **Mount on** column for the **apps** volume, and specify **D**:
 - d. Click on the field under the **Mount on** column for the **SQL_ISO** volume, and specify **E**:
 - e. Click on the field under the **Mount on** column for the **SDM_ISO** volume, and specify **F**:
 - f. Click the **Read-only** and **Share** buttons under the **Options** column for both the **SDM_ISO** and **SQL_ISO** volumes.
5. Click the **Properties** tab and add the following properties by clicking the **Add** button, and then clicking in fields to change their values:

Name	Type	Default	Mandatory?
sql2008r2_prod_key**	String	(empty)	No
SDM_service_name	String	(empty)	Yes*

* To make a property mandatory, click the **Mandatory** button under the **Options** column.

**The sql2008r2_prod_key is used to license Microsoft SQL Server 2008 R2 client in this example. Depending on how the license agreement is structured, the client may or may not need to be licensed.

- Click the **Resources** tab, and modify the resources as follow:

Resource	Min	Max	Default
CPU (num)	2	4	2
Memory (bytes)	4G	8G	4G
Bandwidth (bits/sec)	100M	200M	100M

- Click the **OK** button in the Class Definition window.
- When prompted to save the application, click **OK**.

SQL0864S Singleton

Next, we are going to modify the SQL0864S singleton to accommodate a CA Service Desk Manager r12.6 database server that will contain the CA MDB. Since we need this to be re-usable (as it will eventually be stored in a catalog) and we need the data to be persisted (as new data will be stored on data volume) we are going to create a separate appliance volume for the singleton which will contain the CA MDB data. This will allow us to store this appliance in a catalog. End users will be expected to branch this appliance once they drag the appliance onto their canvas of their new application.

Perform the following steps.

- Right-click the **SQL0864S** singleton, and click **Modify Boundary** in the context menu.
- Click the **General** tab and make the following changes:
 - Change the *Name* to **SDM_MDB_12_6_S**.
 - Change the *Instance Name Template* to **MDB**.
 - Change the *Category* to **CA_Service_Desk_Manager_12_6**.

- Change the *Description* to: **SQL Server Appliance for CA Service Desk Manager r12.6 which also contains the CA MDB - based on Windows Server 2008 Standard Edition R2 64-bit and SQL Server 2008 Standard R2 SP1; CA build (v1.0.0-1)**
- Change the *Color* to **Green**.
- Change the *Size* to **Large**.
- Verify that the *OS Icon* is set to **Windows**.

3. Click the **Volumes** tab and perform the following:

- a. Select the **data** volume of type **Placeholder** and click the **Delete** button.
- b. Add the following volumes by clicking the **Add** button, specifying the information below, and clicking **Next**:

Type	Name	Size	Filesystem
Instantiable	data	20G	ntfs08
Placeholder	SDM_ISO	N/A	N/A

- c. When finished, click on the field under the **Mount on** column for the **data** volume, and specify **D:**
- d. Click on the field under the **Mount on** column for the **SDM_ISO** volume, and specify **E:**
- e. Click the **Read-only** and **Share** buttons under the **Options** column for the **SDM_ISO** volume.

Note: The original SQL0864S appliance was built assuming that the database data would be stored on an application user volume. In this example, the existing data placeholder volume will be deleted, and a new appliance volume will be created. This will enable an end user to drag and drop this new appliance onto a canvas and the appliance will be ready to start up without requiring creating of any more volumes.

4. Click the **Properties** tab and delete the following properties by selecting the property, and clicking the **Delete** button:

- user_db_name
- user_login
- user_password

Note: Because the CA MDB will be installed these properties are no longer needed.

- Click the **Resources** tab, and modify the resources as follows:

Resource	Min	Max	Default
CPU (num)	2	8	2
Memory (bytes)	4G	64G	4G
Bandwidth (bits/sec)	100M	200M	100M

- Click the **OK** button in the Class Definition window.
- When prompted to save the application, click **OK**.

Step 5: Add Properties to the Application Boundary

Our application now consists of the following:

- **SDM_IN_12_6** singleton based on the standard IN gateway appliance
- **SDM_PRI_12_6_S** singleton based on the WIN0864S appliance
- **SDM_MDB_12_6_S** singleton based on the SQL0864S appliance
- **MON** appliance

Next, we will define application properties which will configure all of the appliances contained in the application. To add properties to the application boundary, perform the following steps:

- In the Application Editor window, click the **Application** menu, and click the **Modify Boundary** menu item.
- Click the **Properties** tab and add the following properties by clicking the **Add** button, click in the fields, and type in the following values:

Name	Used to Configure...	Mandatory?*
win2008r2_prod_key	<i>All Windows appliances</i>	No
kms_server_port	<i>All Windows appliances</i>	No
sql2008r2_prod_key	<i>All appliances with any Microsoft SQL Server components installed</i>	No
SDM_service_name	SDM_PRI_12_6_S	Yes
SDM_IN_ip_addr	SDM_IN_12_6	Yes

Name	Used to Configure...	Mandatory?*
NET_ip_addr	NET	Yes
NET_dns1	NET	Yes
NET_dns2	NET	No
netmask	SDM_IN_12_6 and NET	Yes
gateway	SDM_IN_12_6 and NET	Yes

*To mark a property as mandatory, click the **Mandatory** button under the **Options** column

3. Click **OK** in the Class Definition window.
4. Click the **Save** button in the toolbar of the Application Editor.

Step 6: Add the Application User Volumes for the MON Appliance

We are now going to create application user volumes that will contain the images for CA Service Desk Manager r12.6 and Microsoft SQL Server 2008 R2. Before the volumes for the images can be created, however, the images must be imported into the grid controller's /vol/_impex directory.

Import ISO Images

The CA Service Desk Manager r12.6 image and the Microsoft SQL Server 2008 r2 image must be mounted within the singleton so that the software can be installed. To mount the software, the images must be imported into the application. Perform the steps below to import the images:

1. If the CA Service Desk Manager r12.6 or Microsoft SQL Server 2008 R2 images are not already available in the grid controller's /vol/_impex directory, import them to the controller now.

Note: This requires grid maintainer privileges.

2. Create application user volumes for both images by clicking on the **Grid Shell** button within the toolbar of the Application Editor, and executing the following command for each image:

```
vol import <application_name>:<volume_name> <iso_image_name>
```

For example:

```
vol import CA_Service_Desk_Manager_12_6_Small:SDM_ISO DVD03162415E.iso
```

Note: Do not use a UNC share to perform these steps. For more information on the vol import command see the CA AppLogic Help or the [Command Line Shell Reference](#).

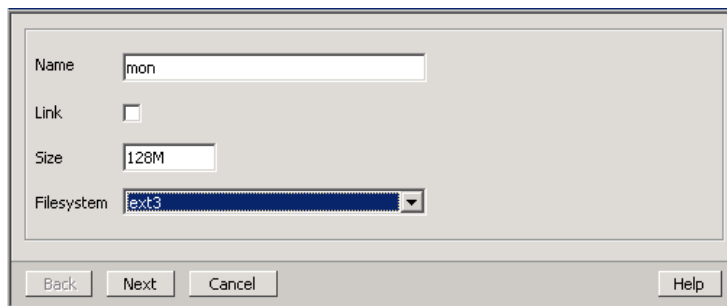
3. Close the Grid Shell.
4. Within the Application Editor, click the **Application** menu, and click **Manage Volumes** in the context menu.
5. Verify that two application user volumes were created: one for the CA Service Desk Manager r12.6 (**SDM_ISO**) image, and one for the Microsoft SQL Server 2008 R2 (**SQL_ISO**) image.

Note: After the software is installed, these volumes can be deleted.

Add an Application User Volume for the MON Appliance.

To add the application user volume to the MON appliance do the following:

1. Within the Manage Volumes dialog, click **Add**.
2. Specify the following information and click **Next** when done:
 - Name: **mon**
 - Size: **128M**
 - Filesystem: **ext3** (since this is a Linux device)



3. Click the **Close** button when done.
4. Click the **Done** button in the Manage Volumes window.
5. Click the **Save** button in the Application Editor.

Step 7: Modify Instance Settings

Now that the class boundaries have been modified, we need to set the Start Order, property values, and user volumes for the individual appliance instances. To set these instance settings perform the following steps in the following sections:

SDM_IN_12_6

For the purposes of this example the SDM_PRI_12_6_S singleton will be configured to ping the SDM_IN_12_6 gateway upon startup, therefore the SDM_IN_12_6 gateway appliance needs to start *before* the SDM_PRI_12_6_S singleton. We also need to redirect some of the appliance settings to the application boundary properties.

To do this, perform the following steps:

1. Within the Application Editor, right-click the SDM_IN_12_6 gateway singleton and click **Attributes** in the context menu.
2. On the **Attributes** tab, perform the following:
3. Change the *Name* to **SDM_IN**.
4. Change the *Start Order* value to **10**.
5. Click the **Property Values** tab to redirect the following appliance properties to the designated application boundary properties:

Appliance Property	Application Property Value
ip_addr	SDM_IN_ip_addr
netmask	netmask
gateway	gateway

To do this:

- a. Click the **Redirect to assembly** button for the ip_addr appliance property.
 - b. Click the field under the **Value** column and select the SDM_IN_ip_addr application property from the drop down list.
 - c. Repeat steps a and b for the remaining appliance properties in the table.
6. Click the **OK** button in the Instance Settings window.
 7. Click the **Save** button in the toolbar of the Application Editor.

SDM_PRI_12_6_S

Upon startup the SDM_PRI_12_6_S will be configured to ping the SDM_IN_12_6 and to attempt to license Microsoft Windows Server and Microsoft SQL Server – which requires access to the external network through the NET gateway. In addition, CA Service Desk Manager r12.6 needs the CA MDB to be available upon startup. Therefore, the SDM_PRI_12_6_S must be configured to start after both the SDM_IN_12_6 and NET gateway appliances, and after the SDM_MDB_12_6_S singleton.

In this step we are also going to add the SDM_ISO application volume and redirect several appliance level properties to point to their application boundary counterparts.

To do this, perform the following steps:

1. Within the Application Editor, right-click the **SDM_PRI_12_6_S** singleton and click **Attributes** in the context menu.

2. On the **Attributes** tab, do the following:
 - Change the *Name* to **PRI**.
 - Change the *Start Order* value to **30**
3. Click the **User Volumes** tab and do the following:
 - a. Click the field under the **App Volume** column for the **SQL_ISO** volume, and select **SQL_ISO** from the drop-down list.
 - b. Repeat step 3a to assign the **SDM_ISO** application user volume, to the **SDM_ISO** singleton volume.
4. Click the **Property Values** tab redirect the following appliance properties to the designated application boundary properties:

Appliance Property	Application Property Value
win2008r_prod_key	win2008r2_prod_key
kms_server_port	kms_server_port
sql2008r2_prod_key	sql2008r2_prod_key
SDM_service_name	SDM_service_name

To do this:

- a. Click the **Redirect to assembly** button for the win2008r_prod_key appliance property.
 - b. Click the field under the **Value** column and select the win2008r2_prod_key application property from the drop down list.
 - c. Repeat steps a and b for the remaining appliance properties in the table.
5. Click the **OK** button in the Instance Settings window.
 6. Click the **Save** button in the toolbar of the Application Editor.

SDM_MDB_12_6_S

We have already determined that the SDM_MDB_12_6_S singleton needs to start *before* the SDM_PRI_12_6_S singleton because the CA Service Desk Manager Primary Server requires access to the MDB upon startup. In addition, the SDM_MDB_12_6_S singleton will be configured to attempt to license Microsoft Windows Server and Microsoft SQL Server upon startup – which requires access to the external network through the NET gateway. As a result, the start order for the SDM_MDB_12_6_S singleton must be *higher* than the NET and *lower* than the SDM_PRI_12_6_S.

We are also going to add the SDM_ISO user volume and redirect specific appliance level properties to their application level counterparts. To do this, perform the following steps:

1. Within the Application Editor, right-click the **SDM_MDB_12_6_S** singleton and click **Attributes** in the context menu.
2. On the **Attributes** tab, do the following:
 - Change the *Name* to **MDB**.
 - Change the *Start Order* value to **20**.
3. Click the **User Volumes** tab, then click the field under the **App Volume** column for the **SDM_ISO** volume, and select **SDM_ISO** from the drop-down list.
4. Click the **Property Values** tab redirect the following appliance properties to the designated application boundary properties:

Appliance Property	Application Property Value
win2008r2_prod_key	win2008r2_prod_key
kms_server_port	kms_server_port
sql2008r2_prod_key	sql2008r2_prod_key

To do this:

- a. Click the **Redirect to assembly** button for the win2008r2_prod_key appliance property.
- b. Click the field under the **Value** column and select the win2008r2_prod_key application property from the drop down list.
- c. Repeat steps a and b for the remaining appliance properties in the table.

Note: For the purposes of this example we are not setting the sa_password property because the sa user credentials are not needed for the install of the CA MDB since the installer uses trusted authentication to create the CA MDB. However, if you want to set the sa_password property, now would be the time to do it.

5. Click the **OK** button in the Instance Settings window.
6. Click the **Save** button in the toolbar of the Application Editor

NET

As we have already determine the NET gateway appliance must be started before both the SDM_PRI_12_6_S and the SDM_MDB_12_6_S singletons because both of those appliances need to connect to the external network to license Microsoft Windows Server and Microsoft SQL Server upon startup. In addition to configuring the start order for the NET appliance we are also going to modify the resources and redirect certain appliance properties to the application boundary. To do this, perform the following steps:

1. Within the Application Editor, right-click the NET gateway appliance and click **Attributes** in the context menu.
2. On the **Attributes** tab change the *Start Order* value to **10**.
Note that this is the same value as the SDM_IN_12_6 appliance which will start at the same time.
3. On the **Resources** tab change the minimum and maximum values for **Bandwidth (bits/sec)** to **200M** and **500M**, respectively.
4. Click the **Property Values** tab redirect the following appliance properties to the designated application boundary properties:

Appliance Property	Application Property Value
ip_addr	NET_ip_addr
netmask	netmask
gateway	gateway
dns1	NET_dns1
dns2	NET_dns2

To do this:

- a. Click the **Redirect to assembly** button for the ip_addr appliance property.
 - b. Click the field under the **Value** column and select the NET_IN_ip_addr application property from the drop down list.
 - c. Repeat steps a and b for the remaining appliance properties in the table.
5. Click the **OK** button in the Instance Settings window.
 6. Click the **Save** button in the toolbar of the Application Editor.

MON

Since the MON appliance is not dependent on any other appliance/singleton, nor is any other appliance/singleton dependent on it we will configure it to start first, along with the SDM_IN_12_6 and NET gateway appliances. We are also going to add the mon user volume.

To do this, perform the following steps:

1. Within the Application Editor, right-click the **MON** appliance and click **Attributes** in the context menu.
2. On the **Attributes** tab change the *Start Order* value to **10** (the same value specific for the SDM_IN-12_6 and NET appliances).
3. Click the **User Volumes**, then click the field under the **App Volume** column, and select the appropriate Application User Volume from the drop-down list:

Name	App Volume
data	mon

4. Click the **OK** button in the Instance Settings window.
5. Click the **Save** button in the toolbar of the Application Editor.
6. Close the Application Editor.

Step 8: Configure the Application

Before the application can be started, we need to configure it by supplying values to the properties that we defined on the application boundary. To do this, perform the following steps:

1. In the CA AppLogic GUI, right-click the application, and click the **Configure** menu item in the context menu.
2. On the **General** tab, provide a meaningful description in the *Description* field. For example: "CA Service Desk Manager r12.6 application"
3. On the **Property Values** tab modify values for the following properties as listed in the table below:

Property Name	Property Value
SDM_service_name	Specify a hostname or ip address. If specifying a host name, it must resolve to the value that will be specified for the property SDM_IN_ip_addr.
SDM_IN_ip_addr	Specify an ip address for this gateway appliance. If a

Property Name	Property Value
	hostname was specified for <code>SDM_service_name</code> , then that value must resolve to this ip address. This value can be obtained from the Grid Controller.
NET_ip_addr	Specify an ip address that will be assigned to the NET gateway appliance. This value can be obtained from the Grid Controller.
netmask	Specify the network mask which can be obtained from the Grid Controller.
gateway	Specify the gateway which can be obtained from the Grid Controller.
dns1	Specify the ip address of the primary DNS server to use. This value can be obtained from the Grid Controller.
dns2	Optionally specify the ip address of the secondary DNS server to use. This value can be obtained from the Grid Controller.

Note: Values for the `win2008r2_prod_key`, `kms_server_port` and `sql2008r2_prod_key` properties should be left blank because we are not licensing the Microsoft Windows or Microsoft SQL Server installations at this point.

4. Click **OK** in the Application Configuration window.

The application will automatically save.

Step 9: Start the Application

We are now ready to install the software on the singletons in the application. First, however, we need to start the application. To do this:

1. In the CA AppLogic GUI, on the Applications tab, right-click on the application, and click **Start** in the context menu.
2. Click the application to open the Application Editor.

Step 10: Login to the Singletons

Next, we need to log into the singletons. If you enabled Remote Desktop on the Windows singletons (see [Enable Remote Desktop](#) for more details) you have several options for logging in.

Login through Remote Desktop

Since we enabled Remote Desktop and opened port 3389 on the SDM_IN_12_6 gateway appliance in our example we can use Remote Desktop to login to the singletons. This option provides a performance improvement over the graphical console.

- To log into the **SDM_PRI_12_6_S** singleton, connect to the ip address that was specified in the application property **SDM_IN_ip_addr**.
- To log into the **SDM_MDB_12_6_S** singleton, first Remote Desktop into the **SDM_PRI_12_6_S** singleton. Then, from within the **SDM_PRI_12_6_S** singleton, Remote Desktop into the **SDM_MDB_12_6_S** singleton using the out terminal of the **SDM_PRI_12_6_S** that connects to the **SDM_MDB_12_6_S** singleton. For the purposes of our example we will connect using the **mdb** terminal.

Graphical Console

If Remote Desktop is not enabled, you can login using the graphic console by doing the following:

1. Click the application within the CA AppLogic GUI to open the Application Editor.
2. Click the singleton to log into, and click the **Login (graphic)** button within the toolbar of the Application Editor.
3. Provide the appropriate Windows credentials to log in.

If the password needs to be reset, with the same singleton selected within the Application Editor, click the **Login (ssh)** button within the toolbar, and execute the following command:

```
net user administrator <new_password>
```

Step 11: Prepare Windows to install CA Service Desk Manager r12.6

Before we can install CA Service Desk Manager r12.6 on the SDM_PRI_12_6_S and the SDM_MDB_12_6_S singletons we need to verify that the user volumes were properly connected and that the computer names are correct.

Verify that the Volumes were Created

At this point in the exercise new volumes should have been added to both the SDM_PRI_12_6_S and SDM_MDB_12_6_S singletons. To verify that these volumes have been created, do the following (for both appliances!):

1. Click **Start > Computer**.

2. Verify the new volume was created as the D drive.

If you do **not** see either a hard disk, or a CD Drive, do the following:

1. With the graphical console open, and the Desktop showing, click the **Server Manager** button in the taskbar on the desktop.
2. Expand the **Storage** node and select **Disk Management**.
3. Right-click the partition that does not appear and click **Change Drive Letter and Paths...**
4. Click the **Add...** button.
5. Select the drive letter from the drop-down list you assigned the volume when modifying the boundary of the singleton, and click **OK**.
6. Repeat steps 3-5 as necessary for each partition that did not appear when clicking **Computer**.
7. Close the Server Manager window.
8. Verify the drives appears now.

Verify the Hostname has been Changed

When we configured the WIN2008R64S base Windows Server 2008 r2 appliance (upon which the SDM_PRI_12_6_S singleton was based) we opted to disable generation of host names (see [Disabling Auto Host Name Generation](#)). This step was optional, however, depending on whether or not you chose to perform it, you may need to manually change the hostname. To do this:

1. Click **Start > Control Panel** to open Windows Control Panel.
2. Click the **System and Security** link.
3. Click the **System** link.
4. Verify that the **Computer name** under the **Computer name, domain, and workgroup settings** section reflects the instance name that was given to the singleton in the Application Editor. For example, we used the following:

Appliance	Computer name
SDM_PRI_12_6_S	PRI
SDM_MDB_12_6_S	MDB

If the computer name is **not** the same as the instance name you can do the following:

1. With the System window open, click the **Change settings** link under the **Computer name, domain, and workgroup settings** section.
2. Click the **Change...** button on the **Computer Name** tab.

3. Change the **Computer name** to the appropriate instance name, and click **OK**.
4. Follow the prompts to close the dialogs, and when prompted to restart now or later, click the **Restart Later** button.
5. Close the graphical console or the Remote Desktop session.
6. Right-click the application within the CA AppLogic GUI and click **Restart** from the context menu.
7. After the application restarts, login to Windows using either the graphical console or Remote Desktop.

Step 12: Install the MDB

Since the CA Service Desk Manager Primary Server requires the MDB, we are going to install that first on the SDM_MDB_12_6_S singleton by doing the following:

1. Login in to the **SDM_MDB_12_6_S** singleton
2. Execute **setup.exe** from the root of the CA Service Desk Manager r12.6 install image
3. Select the language and click **Select Language**.
4. Click on **Product Installs**
5. Click on the **CA MDB** link
The Welcome screen appears.
6. Click **Next**
7. On the **CA MDB Location** screen, install the CA MDB to **D:\CA\SC\Mdb** and click **Next**. Proceed to creating the folder when you are notified that the folder does not exist.
8. Select **Microsoft SQL Server** from the **Database Type** drop-down list, and specify a password for the **Service Desk** user. Click **Next**.
Note: Verify that the **Database Server Name** matches the Instance Name of the singleton. If it does not, try using **in** as the **Database Server Name**. This is the name of the input terminal for the singleton, which is also an alias for this singleton.
9. Click the **Space** tab to confirm sufficient free space is available
10. Click **Finish** to install the MDB, and wait for the install of the CA MDB to finish.

Step 13: Install the Microsoft SQL Server Client

Next, install the Microsoft SQL Server 2008 r2 client, along with the Microsoft SQL Server Management Studio application, on the boot drive of the SDM_PRI_12_6_S singleton. To do this, follow the standard installation prompts

Note: Depending on how it will be used you may or may not need to license the SQL Client. In this example, when the **Product Key** page is shown, we select the option **Specify a free edition**, and choose **Evaluation** from the drop-down list.

Step 14: Install CA Service Desk Manager

We are now ready to install CA Service Desk Manager on the SDM_PRI_12_6_S singleton. To do this:

Note: The installation steps provided below are designed to demonstrate the capability of the CA AppLogic system rather than a full and proper CA Service Desk installation. For this reason we have opted not to include or implement many of the options and best practices associated with CA Service Desk.

1. Execute **setup.exe** from the root of the CA Service Desk install image
2. Select a language, and click the **Select Language** button.
3. Click on **Product Installs**
4. Click on the **CA Service Desk Manager** link
5. Read the Introduction screen and click **Next** to continue
6. Accept the license agreement by scrolling to the bottom of the agreement, and clicking **Next**.
7. When prompted to choose a folder to install CA Service Desk Manager to specify the following and click **Next**:

D:\CA\Service Desk Manager
8. When prompted to choose a folder to install the Shared Components to, specify the following and click **Next**:

D:\CA\SC
9. Review the information provided on the screen and click **Install**.

The CA Service Desk Manager installer will complete the installation, and the CA Service Desk Manager Configuration window will appear.
10. On the **General Settings** screen, specify **Primary Server** for the *Configuration Type*, and click **Next**.

Note: By default, the values specified for *Primary Server Node* and *Object Manager Display Name* should both reflect the hostname/instance name of the singleton.
11. On the **System Accounts** screen, specify passwords for both the **ServiceDesk** application user, and the **rhuser**, and click **Next**.

Note: The passwords for both users can be the same, or different.
12. On the **Select Database** screen, select **SQL** from the drop-down list, and click **Next**.
13. On the **MS SQL Database Config** screen, do the following:

- verify that *Load default data* is checked
- specify the terminal name that connects to the `SDM_MDB_12_6_S` singleton (which is **mdb** in our example)
- specify the password for the **ServiceDesk** database user that was used when installing the CA MDB

Then, click **Next**.

Note: When the *SQL Listening Port* is left blank, it uses the default port 1433.

14. On the **Web Interface** screen, verify the *Web Host* is the name of the computer/instance name of the singleton, and verify that **Tomcat Server only** is selected for *Config Type*, and click **Next**.

Note: Port 8080 is the port end users will use to connect to the CA Service Desk Manager application.

15. On the **Visualizer** screen, check the checkbox for *Configure Visualizer*, and click **Next**.

Note: Port 9080 is the port end users will connect to for the CA CMDB Visualizer.

16. On the **Support Automation** screen, select **Main Server** from the drop-down list for *Configuration Type*, verify that wherever the hostname is provided that it also is the instance name of the singleton, and click **Next**.

Note: Port 8070 is the port end users will use to connect to CA Support Automation.

17. On the **Config Options** screen, uncheck *Start service when completed*, and click **Finish**.

Note: For the purposes of this example we are using scripts that will start the service upon startup.

18. Wait for the configuration to complete.
19. Click the **OK** button in the message dialog saying that configuration is complete.
20. Click **Next** when prompted to review the release notes.
21. Click **Next** without launching the CA Service Desk Manager website.
22. Click the **Done** button.
23. Close the CA Service Desk Manager r12.6 Installation Menu.

If any errors appear, review the following files which can be found in the path `D:\CA\Service Desk Manager\log`:

- `configure.log`
- `stdlog.0`

Step 15: Create the Scripts

In our example we are going to use two scripts: one for the Microsoft SQL Server Client, and one for CA Service Desk Manager r12.6.

Microsoft SQL Server Client Script

If you configured Microsoft SQL Server Client to be licensed by end users, you should consider providing a means to automate this. One way to do this is to write a script, which enables you to **provide a standard way for end users to license Microsoft SQL Server Client**. For example, the Microsoft License Key Injection Utility (discussed in [Appendix A](#)) utilizes a new boundary property – `sql2008r2_prod_key` - to inject the appropriate licensing details for the appliance. This script used by this utility is stored in the `C:\appliance\init\mspk` folder and generates log messages to a file stored in the directory `C:\appliance\init\logs` folder. It also includes a common utilities script that is created and stored in the `C:\appliance\init\util` directory. For more information about these scripts, as well as a command that can be called to license Microsoft SQL Server (and the Client), see [Appendix A](#).

If you plan on using scripts to license the Microsoft SQL Server Client, create and implement those scripts now.

CA Service Desk Manager Script

Since we intend to eventually turn this application into a template application (see the section [CA AppLogic Templates](#) for more details) to enable other users to provision their own copy of the application, we are also creating a script for CA Service Desk Manager. This script will execute upon startup, so that when another user starts their own instance of the application, CA Service Desk Manager r12.6 will be ready to use immediately.

Modify CA Service Desk Manager Script

CA Service Desk Manager provides a batch file, `checkDSMhost.bat` (found in the `D:\CA\Service Desk Manager\bin\` folder) which can be executed from an environment, like Cygwin, to reconfigure CA Service Desk Manager so that it can be accessed by an end user when CA Service Desk Manager is behind a firewall. This batch file, calls the `checkSDMhost.pl` perl file (also found in the `D:\CA\Service Desk Manager\bin` folder).

The `checkSDMhost.bat` file accepts an input parameter which we are going to use to pass in the external hostname (using the `SDM_service_name` property) to configure CA Service Desk Manager r12.6. By default, the external name is set to the actual hostname, which cannot be accessed outside of a firewall. Therefore, our first step is to modify the `checkSDMhost.pl` perl script, using an editor like VI. Following is a high level overview of how to do this:

1. First, make a backup copy of the `checkSDMhost.pl` file
2. Open the `checkSDMhost.pl` file using an editor like VI.
3. Locate the sub routine called **getActualHostValues**.

This sub routine sets the external hostname through which CA Service Desk Manager will be accessed.

4. Modify the sub routine to set the `$publicDNSName` variable to the value of the input parameter that was passed in (which represents the external hostname).

For example, you could add the following line to the sub routine:

```
$publicDNSName = "$ARGV[0]";
```

Note: You will need to take into consideration when no input parameter is passed in.

5. Save the file.

Create Scripts to Configure CA Service Desk Manager Upon Startup

Next we need to create a script that will execute the `checkSDMhost.bat` file. The script should include the following major tasks:

1. Check first if the existing public dns name matches the value that was passed into the `SDM_service_name` property.

Note: The existing public dns name can be found in `D:\CA\Service Desk Manager\SDUninstall\installvariables.properties` and can be found in the property **szPublicDNSName**. If the property is not there, that is OK as that means the `checkSDMhost.bat` has not been executed.

If it does **not** match, do the following:

- a. Execute the following command:

```
checkSDMhost.bat $SDM_service_name
```

- b. Start the CA Service Desk Manager r12.6 Windows service.
- c. Install the CA Support Automation options using the following commands which are executed from the `D:\CA\Service Desk Manager\bin` directory:

```
pdm_options_mgr.exe -b -c -a pdm_option.inst -s supportautomation_url -v  
'http://<external_host_name>:8070/SupportAutomation'  
pdm_options_mgr.exe -c -b -a pdm_option.inst -s sa_primary_domsrvr
```

- d. Restart CA Service Desk Manager
2. If the existing public dns name **does** match the value passed into the `SDM_service_name` property, start the CA Service Desk Manager r12.6 Windows service.
 3. Open the `C:\cygwin\etc\sysconfig\applogic_appliance` file using an editor like VI and append a line at the end of the file that executes the script.

For example:

```
/appliance/init/sdm/sdm_init.sh
```

Note: These steps assume that the `C:\appliance\init\SDM` folder already exists or has been created.

4. Save the file when done.

Execute CheckSDMhost and Update a CA Service Desk Manager File

There is a known issue with a file in CA Service Desk Manager that does not get updated when the checkSDMhost.bat command is run - specifically when the current hostname found in the file matches the hostname of the computer. As a workaround, this command will change all of the files to something different than the hostname, and then we will have to manually update those files. Once we do this, then checkSDMhost.bat will update all files accordingly. Perform the following steps.

1. Double-click the **Cygwin** application on the desktop to open it.
2. Execute the following command:

```
checkSDMhost.bat SDM_service_name
```

Note: SDM_service_name is not a substitute for anything else, so actually type out 'SDM_service_name' as the input parameter to the checkSDMhost.bat command.

Note: There is a known issue with a file in CA Service Desk Manager that does not get updated when the checkSDMhost.bat command is run. This specifically occurs when the current hostname found in the file matches the hostname of the computer. As a workaround, this command will change all of the files to something different than the hostname, and then we will have to manually update those files. Once we do this, then checkSDMhost.bat will update all files accordingly.

3. Open the following file in notepad:

```
D:\CA\Service Desk Manager\bopcfg\www\CATALINA_BASE_SA\webapps\SupportAutomation\WEB-INF\classes\config\server.properties
```

4. Find the lines towards the bottom where it reads:

```
...  
URLHost: PRI  
...  
SocketServerHost: PRI
```

5. Replace **PRI** with **SDM_service_name** in those lines, and save the file. Close the file when done.

6. Open the following file in notepad:

```
D:\CA\Service Desk Manager\SDUninstall\installvariables.properties
```

7. Verify the last line of the file reads:

```
szPublicDNSName=SDM_service_name
```

8. If it does, then just close the file; otherwise, change the line so that it does read that way, then save and close the file.

Step 16: Prepare the Singletons to Become Appliances

We are now ready to prepare the modified singletons to be transformed into appliances by being added to a catalog. To do this, perform the following steps on both the `SDM_MDB_12_6_S` and `SDM_PRI_12_6_S` singletons:

1. Review and follow the preparation guidelines used earlier for the Windows 2008 r2 appliance (see [Step 8: Prepare the Singleton to be Added to a Catalog](#)).
2. Either close the graphical console if it is still open, or log off the Remote Desktop session.
3. Right-click the application within the CA AppLogic GUI, and click **Stop** in the context menu.
4. Click the application to open the Application Editor.
5. Right-click the `SDM_PRI_12_6_S` singleton and click **Modify Boundary** in the context menu.
6. Click the **Volumes** tab.
7. Select the following volumes(one at a time), and click the **Delete** button, as they are no longer needed. When prompted to first remove the parameterization, click **OK**:
 - SQL_ISO
 - SDM_ISO
8. Click the **OK** button in the Class Definition window.
9. Click the **Save** button in the toolbar of the Application Editor.
10. Repeat steps 5-9 to delete the `SDM_ISO` volume from the `SDM_MDB_12_6_S` singleton.
11. Click the **Application** menu, and click **Manage Volumes** in the context menu.
12. Select the following volumes(one at a time), and click the **Delete** button, as they are no longer needed. When prompted to delete the volume and its references, click **OK**:
 - SDM_ISO
 - SQL_ISO
13. Click the **Done** button in the Manage Volumes dialog.
14. Click the **Save** button in the toolbar of the Application Editor. Leave the Application Editor open.

Step 17: Store Appliances in the CA AppLogic Catalog

Now that they have been prepared we can store the singletons as appliances in the AppLogic catalog. To do this:

1. Click the **Grid Shell** button in the toolbar of the Application Editor, and execute the following command:

```
cat create <catalog_name>
```

Note: To make the new catalog accessible from other grids, you will need to define it as global by adding a slash ("/") in front of the catalog name.

2. Close and reopen the Application Editor by clicking on the application within the CA AppLogic GUI.

The new catalog should now be listed in the catalog dropdown.

3. Select the new catalog
4. Drag the following singletons from the canvas into the new catalog so that the new catalog will contain all of the appliances necessary for the application:
 - SDM_IN_12_6
 - SDM_PRI_12_6_S
 - SDM_MDB_12_6_S

The following commands can be executed from a Grid Shell to export and import a catalog to another grid.

- Export the catalog using the following command:

```
cat export
```

- Import the catalog using the following command:

```
cat import
```

Step 18: Test the Application and Finalize the Application

Typically, if we were to run the application as is, we would want to enable data persistence by branching both the SDM_MDB_12_6_S and SDM_PRI_12_6_S appliances. However, for purposes of verification, we are just going to test that CA Service Desk Manager is available upon startup. To do this, perform the following steps:

1. Right-click the application within the CA AppLogic GUI, and click **Start** in the context menu. Wait for the application to start as it will take awhile.
2. Open Internet Explorer on a remote computer and navigate to:

http://<SDM_service_name>:8080

Note: Replace <SDM_service_name> with the value that was provided for the application property SDM_service_name when the application was [configured](#).

3. Login as the application user **ServiceDesk**.
4. Click the **Support Automation** tab and click the **Live Assistance** button to make sure the Analyst Console displays.

Note: This requires a java runtime environment (jre) version 1.6 or later to be installed on the client computer.

5. Close the Analyst Console.
6. Click the **Administration** tab.
7. Expand **CA CMDB** on the left.
8. Click on the link for **CI List**.
9. Click the **Create New** button on the right hand side.

This is to test to make sure the CA CMDB features are working.

10. Type **Test** in the *Name* field, and type **Hard Drive** in the *Class* field.
11. Click the **Continue** button.
12. Click the **Save** button.
13. Click the **Asset Viewer** button.
14. Verify the **Unicenter Common Asset View** displays and close the window when done.
15. Click the **CMDBfViewer** button.
16. Verify the **Configuration Item Federation Viewer** displays and close the window when done.
17. Click the **Visualizer** button.

Note: The client computer must have a JRE version 1.6 or later installed, and is recommended to have at least a single 1.0 GHz processor or better with at least 1 GB of RAM.

18. Click the **Create CI** button within the toolbar of the CMDB Visualizer.
19. Verify that the **Create New Configuration Item** window appears and close the window when done.
20. Click the **Create Relation** button within the toolbar of the CMDB Visualizer
21. Verify that the **Create New Configuration Item Relationship** window appears and close the window when done.
22. Close the **CMDB Visualizer**.
23. Close the **Configuration Item Detail** window.
24. Click the **Log Out** link next to the **ServiceDesk** username, and close Internet Explorer.
25. Right-click the CA AppLogic CA Service Desk Manager application, within the CA AppLogic GUI, and click **Stop** in the context menu.

Once we have verified that CA Service Desk Manager can be connected to we can delete any configuration changes that were made, returning the appliances back to their initial states. We can then branch the `SDM_MDB_12_6_S` and the `SDM_PRI_12_6_S` appliances to enable data persistence. To do this:

1. With the application selected in the CA AppLogic GUI, click the **Grid Shell** button within the toolbar, and execute the following command:

```
app clean
```

2. Close the Grid Shell when done.
3. Click the application within the CA AppLogic GUI to open the Application Editor.
4. Right-click the `SDM_PRI_12_6_S` appliance, and click **Branch Class** from the context menu.
5. Right-click the **PRI** singleton and click **Modify Boundary** in the context menu.
6. Change the **Name** to `SDM_PRI_12_6_S`. Click the **OK** button when done.
7. When prompted to save the application, click **OK**.
8. Repeat steps 4-7 for the `SDM_MDB_12_6_S` appliance and to change the name back to `SDM_MDB_12_6_S` after the appliance has been branched.

The application is now done. In the next section we are going to demonstrate how to convert the application into a *template application* to maximize re-usability.

Chapter 9: Advanced Application Configuration Tips

This chapter looks at some additional, more advanced options that you can use to leverage the full functionality of CA AppLogic. The following topics are discussed:

- [Creating Application Templates](#)
- [Creating Assemblies](#)
- [Creating an Application with an External SQL Server](#)

This is not an inclusive list of advanced configuration options and this section may be expanded to explore other options in future editions of this document.

Before beginning any advanced appliance creation tasks it is highly recommended that you familiarize yourself with the relevant supporting materials, such as the product documentation, which can provide more in depth detail. Following is a list of suggested reading:

- The CA AppLogic Application Model
http://doc.3tera.com/AppLogic30/en/User_Guide/1475591.html
- The Application Descriptor Language (ADL) Reference Guide
http://doc.3tera.com/AppLogic30/en/Adl_Ref/index.htm
- The AppLogic Appliance Kit
http://doc.3tera.com/AppLogic30/en/Developer_Guide/ApplianceKit.html

The topics discussed in these documents are fundamental to the development of applications and appliances in CA AppLogic. Much of the information provided in this chapter - and in the document as a whole - is procedural in nature, and it is not the author's intent to distract from that overall flow. It is important, however, that you fully understand the concepts applied in the procedures so that they can be applied to other examples.

Application Templates

Once you have created an application, there may be some cases where you may need to deploy that application multiple times. For example, as an MSP you may want to provision the CA Service Desk Manager application we just created for several different customers. This can be accomplished by saving the application as a *template*, and then, provisioning a separate copy of the application for each customer.

Create an Application Template

For the sake of simplicity we are going to demonstrate the application template creation process using the CA Service Desk Manager r12.6 application that was just created. However, before saving that application as an application template there are certain modifications we will need to make to it to enable us to redeploy the application later. To make these changes, perform the following steps:

1. Within the CA AppLogic GUI, right-click the CA Service Desk Manager application that was just created, and click **Configure**.
2. On the **General** tab, verify that a meaningful *Description* is specified.
3. Click the **Resources** tab.
4. Verify that the total values for the *Min*, *Max* and *Default* number of resources defined for the application match what is expected from the internal appliances and singletons.

Note: After configuring the resources for each of the singletons and appliances within the application, CA AppLogic should aggregate those resources at the application level. These values can be modified, as needed, at deployment, provided they are set when the template is created.

5. Click the **Property Values** tab.
6. Verify that all desired application properties have been defined (see the [Step 5: Add Properties to the Application Boundary](#) section for more details on defining application properties).

The values for these properties will be inherited by one or more appliances and singletons.

7. Click the **Reset All** button to clear all values.

This enables the end user who is provisioning a copy of this application to specify their own values.

8. Finally, click the **General** tab again, and check the **Template** checkbox.

This marks this application as a template and makes it provisionable.

9. Click the **OK** button in the Application Configuration dialog to save the changes.

10. Verify in the CA AppLogic GUI that the application name has been appended with **(template)**.

Provision and Deploying an Application from an Application Template

Since we saved the application as a template, we can reuse it as often as we like by just provisioning and deploying it. To do this, perform the following steps:

1. Right-click the CA Service Desk Manager template application within the CA AppLogic GUI, and click **Provision** in the context menu.

2. On the **General** page, optionally specify a new name and description and click **Next** when done.
Note: By default, CA AppLogic will increment the application name to avoid duplicates.
3. On the **Configure Resources** page adjust resource allocations (or leave the defaults as is) and click **Next**.
Note: By default, the resources should be aggregated from all of the appliances and singletons within the application, so accepting the default values would be sufficient.
4. On the **Configure Properties** page, enter the desired property values by clicking in the field under the **Value** column and typing in the value. Click **Next** when done.
Note: For more information on each of the properties we defined for the CA Service Desk Manager application, see the section [Step 8: Configure the Application](#).
5. On the **Finalizing...** page, optionally check either of the following checkboxes for the following reasons:
 - **Start application after provisioning** – If it is desired to start the application after provisioning.
 - **Use file-system level copies when copying the volumes of the new application** - May be helpful if you have large volumes with little data.
 - **When creating new volumes for the application, prefill all blocks in the volumes** - If it is desired to use file-system level copies, then you can also prefill all blocks in the volumes by checking this checkbox. This could cause provisioning to take longer.
6. Click **Next** on the **Finalizing...** page.

The application will now be provisioned. Depending on the size and complexity of the application, provisioning may take 30 minutes or more since CA AppLogic recreates all of the volumes and, optionally, starts the application.

When provisioning is complete, if the option **Start application after provisioning** was not checked, optionally start the application at this time by right-clicking the application within the CA AppLogic GUI, and clicking **Start** in the context menu.

Assemblies

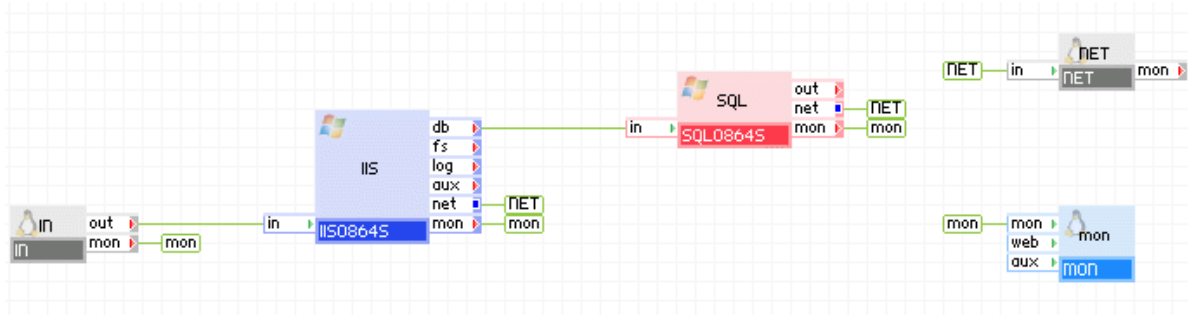
Thus far we have created a Windows appliance and then used it as the basis for creating other appliances, including a Microsoft IIS appliance and the CA Service Desk Manager r12.6 primary server appliance. Suppose, now, that these applications are elements of a package that you want to provide to your customer. You *could* set up a grid and provision each of the applications separately, however, remember that one of the goals of cloud computing is to bring solutions to your customers as *rapidly* as possible. One way to do this is to provide your customers with a "data center ops" application that contains all of these applications in a single package. CA AppLogic enables you to do this through the use of *assemblies*.

Application templates enable you to replicate an application on the grid. What assemblies do is enable you to embed those applications to provide a pre-packaged, multi-functional solution.

Designing Assemblies

The process of creating assemblies is not very different from the process we used to create applications so far. Therefore, for the sake of simplicity, we are going to use our [Windows IIS Application Example](#) to illustrate the differences.

Here you can see the IIS Application:



Now, let us walk through how to build the exact same application as an assembly.

Note: The steps provided include only the high level details. For more detailed information on certain steps, such as the addition of application user volumes, refer to the earlier sections of the guide.

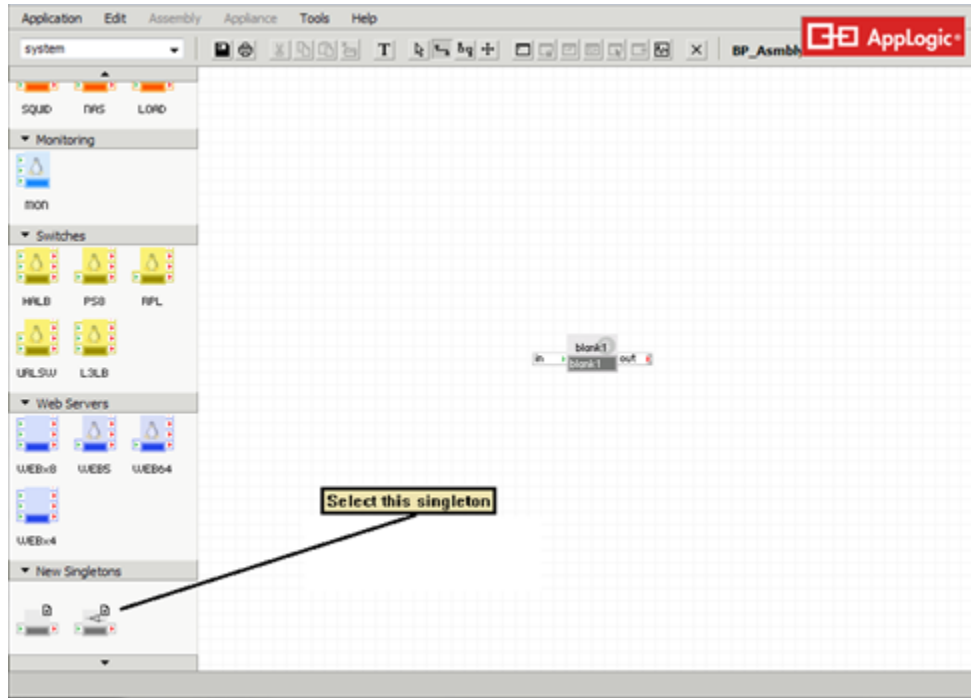
Step 1: Installing the Application

The first step is to create a new application. To do this:

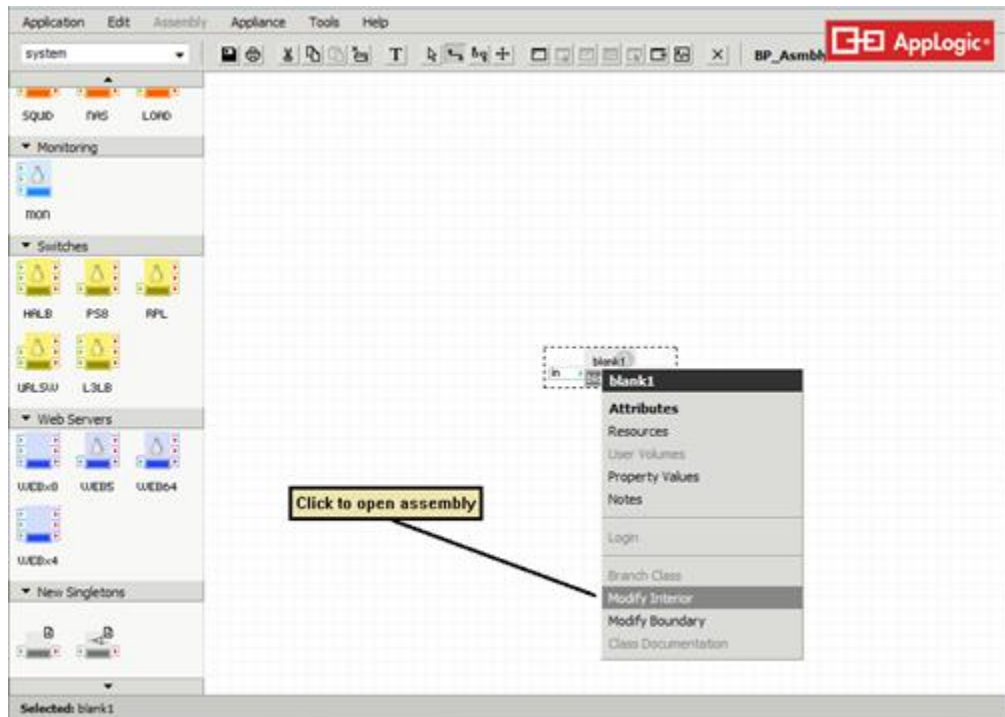
1. Create a new application in AppLogic by following the steps provided earlier in this guide.
2. Open the Application Editor by clicking on the application.

3. Locate the **New Singletons** category in the **system** catalog.

In the following example you can see that there are two singletons. Drag the singleton with the small network symbol on the icon onto the canvas. This will represent our IIS application.



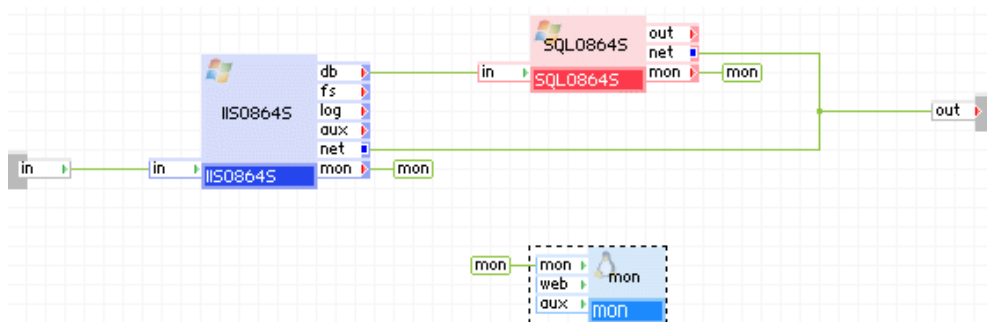
4. To open the assembly, right click on it and click **Modify Interior** in the context menu.



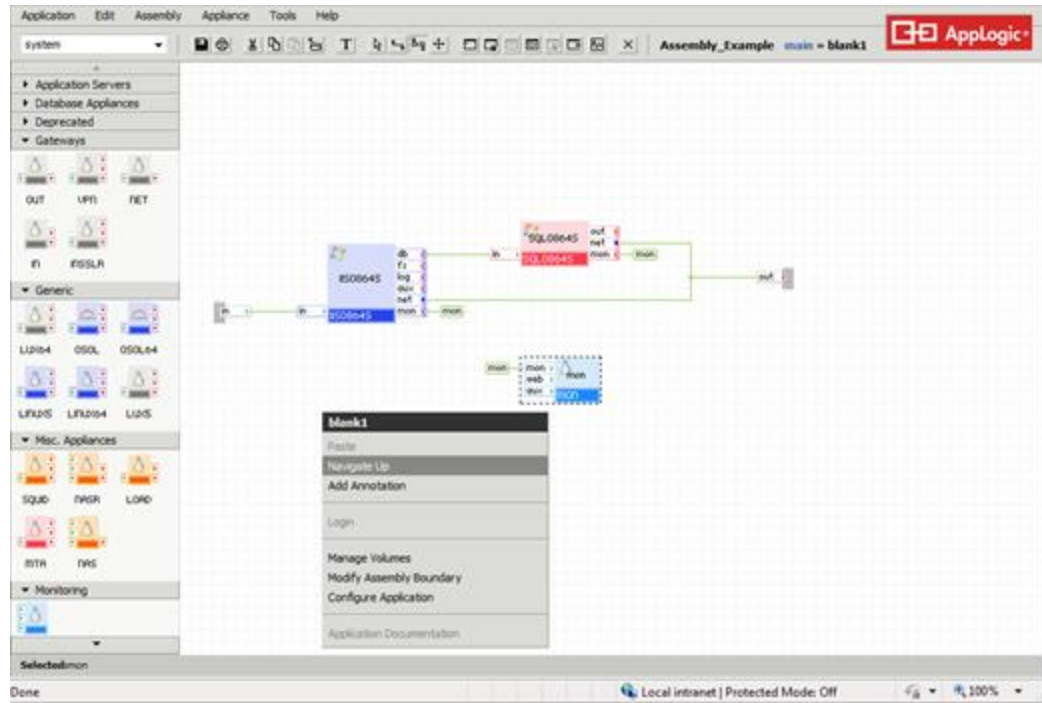
Note that you now have two objects - **in** and **out**. These are gateways into and out of the assembly. Because an assembly has built-in gateways, we won't need to add conventional gateways, such as IN and NET within the assembly. If you need to filter and redirect traffic within the assembly, you can use a PS8 switch.

5. Drag and drop the following appliances onto the canvas:
 - IIS0864S
 - SQL0864S
 - MON
6. Connect them as follows:
 - Connect the **in** gateway to the **in** terminal of the **IIS0864S** appliance
 - Connect the **net** terminal of the **IIS0864S** appliance to the **out** gateway
 - Connect the **net** terminal of the **SQL0864S** appliance to the **out** gateway.
7. Click the **Save** button in the toolbar of the Application Editor.
8. Add the necessary Application User Volumes, by clicking the **Application** menu in the Application Editor, and clicking **Manage Volumes** in the context menu.
9. Once done, configure the appliances to use these volumes.

Your assembly should now look something like the following:



10. Modify the start order of the appliances so that IIS0864S starts *after* SQL0864S.
11. Now that the appliances have been set within the assembly, right click anywhere on the canvas and click "Navigate Up" to close the assembly.



Step 2: Configure the Assembly Class

Next we need to configure the assembly *singleton* to prepare it to become an assembly *class*. To do this:

1. Right click on the assembly icon and click **Modify Boundary**.
2. Provide a *Name* for the assembly and specify a *Category* where it should be stored within a catalog.
3. Change other fields to suit your requirements.
For example, we've changed the *Color* to **Yellow** and made it a **Medium**-sized icon.
4. Click **OK** to close the Class Definition window.
5. When prompted to save the application, click **OK**.
6. Right click the assembly icon again and click **Attributes**.
7. Provide a *Name* for the assembly and click the **OK** button.

The assembly now looks like this:



8. Click the **Save** button in the toolbar of the Application Editor.

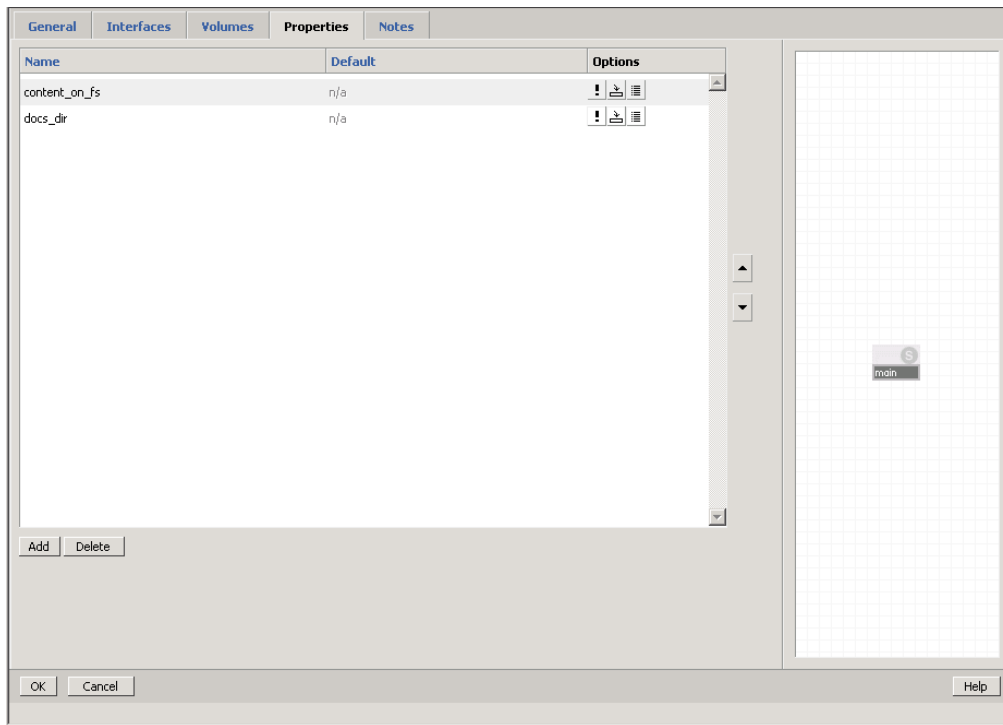
Step 3: Property Management

Properties can be defined at the application boundary level and their values can be passed into the appliances, through the assembly. For example, the IIS web server includes two key properties that are used to define the web server: "docs_dir" and "content_on_fs". The docs_dir property defines where the web site's files reside on the content volume and the content_on_fs defines where the content volume is located (for more information on creating the IIS Web Server application see [Windows IIS Application Example](#)). These properties are essential to properly configuring the web site and, through CA AppLogic, we can define them at the application boundary level by representing each value as a boundary property.

1. With the Application Editor opened, click the **Application** menu, and click **Modify Boundary** within the context menu.
2. Click the **Properties** tab.
3. Add the following properties by clicking the **Add** button, and clicking the field under the **Name** column to type in the name of the property. Click **OK** when done:

Name	Mandatory?
content_on_fs	No
docs_dir	No

Here you can see an example of the dialog used for this:



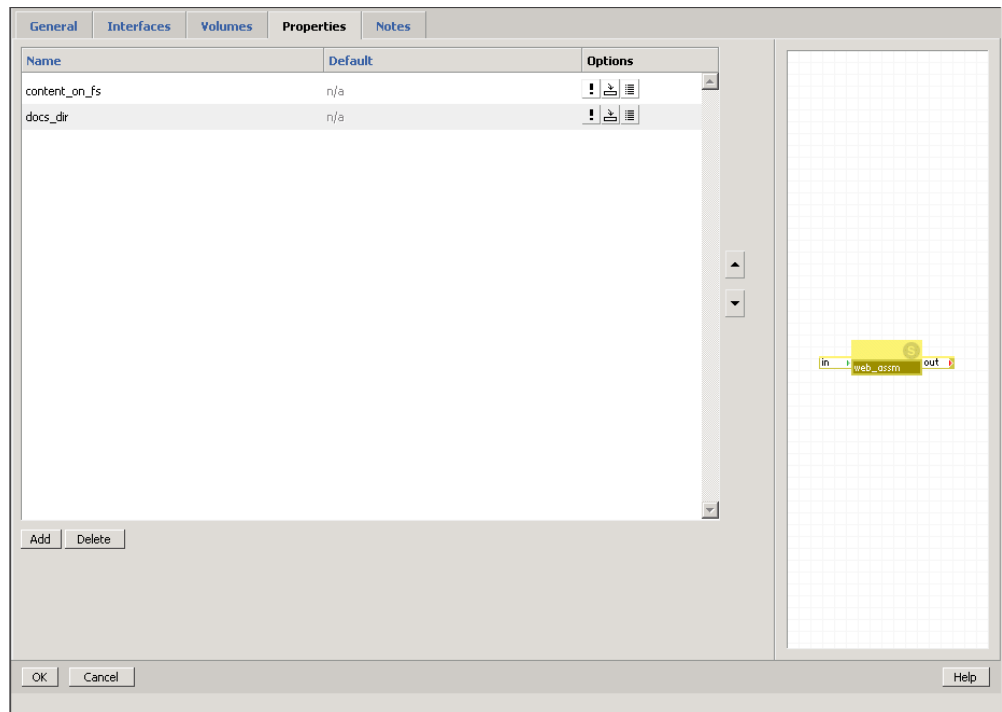
Note: For the purposes of our example we aren't defining docs_dir as mandatory since, when this value is left blank, the directory defaults to residing in the root of the content volume.

4. Click the **Save** button in the toolbar of the Application Editor.

Next, we add properties to the assembly.

1. Right-click the assembly, and click **Modify Boundary**.
2. Click the **Properties** tab.
3. Add the following properties by clicking the **Add** button and clicking the field under the **Name** column to type in the property name. Click **OK** when done.

Name	Mandatory?
content_on_fs	No
docs_dir	No

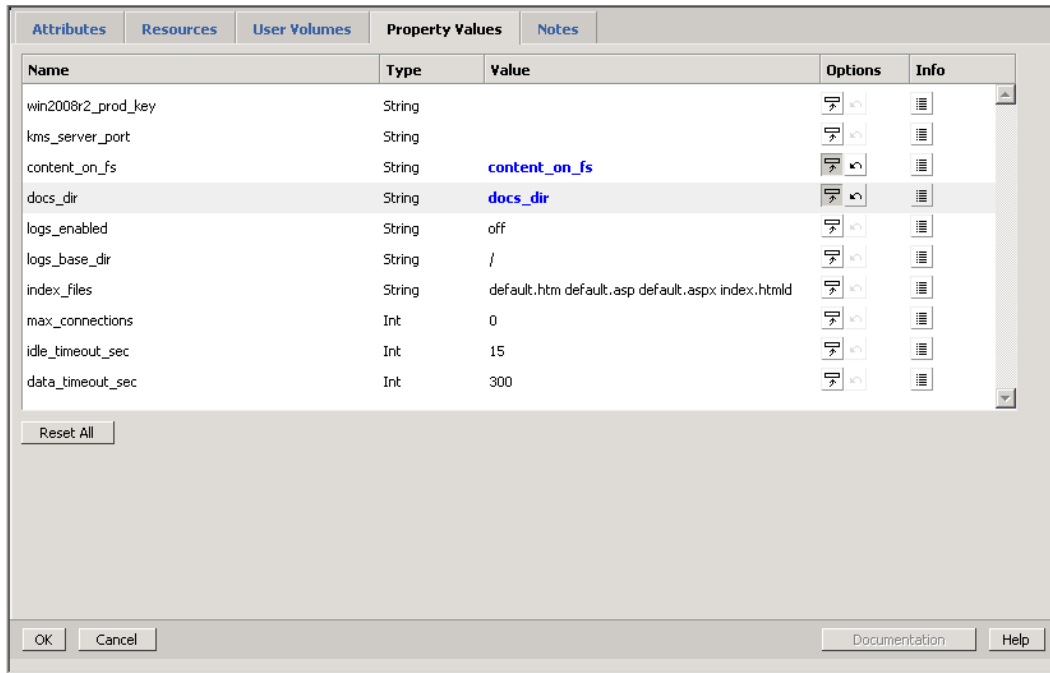


4. Click the **Save** button in the toolbar of the Application Editor.

Next, we assign the assembly properties in the IIS server appliance.

1. Right-click the assembly, and click **Modify Interior** in the context menu.
2. Right-click the IIS0864S appliance, and click **Property Values** from the context menu.

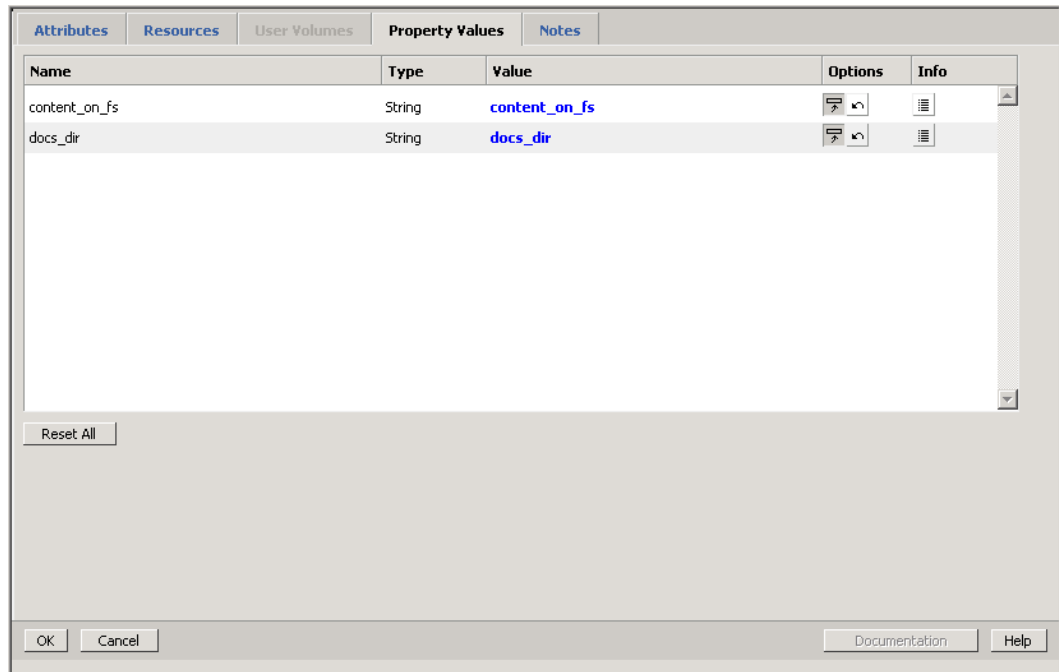
3. Click the **Redirect to assembly** button under the **Options** column for the **content_on_fs** property.
4. Click in the field under the **Value** column, and select the assembly property **content_on_fs**.
5. Repeat steps 3-4 for the docs_dir property to have to appliance property docs_dir inherit the value of the assembly property docs_dir
6. Click **OK** when done.



7. Click the **Save** button in the toolbar of the Application Editor.

Next we will assign the application level properties to the assembly properties.

1. Right-click anywhere in the canvas of the Application Editor, and click **Navigate Up**.
2. Right-click the assembly, and click **Property Values** from the context menu.
3. Repeat steps 3-4 in the previous section to assign the values of the application properties, **content_on_fs** and **docs_dir**, to the assembly properties, **content_on_fs** and **docs_dir**, respectively.
4. Click **OK** when done.

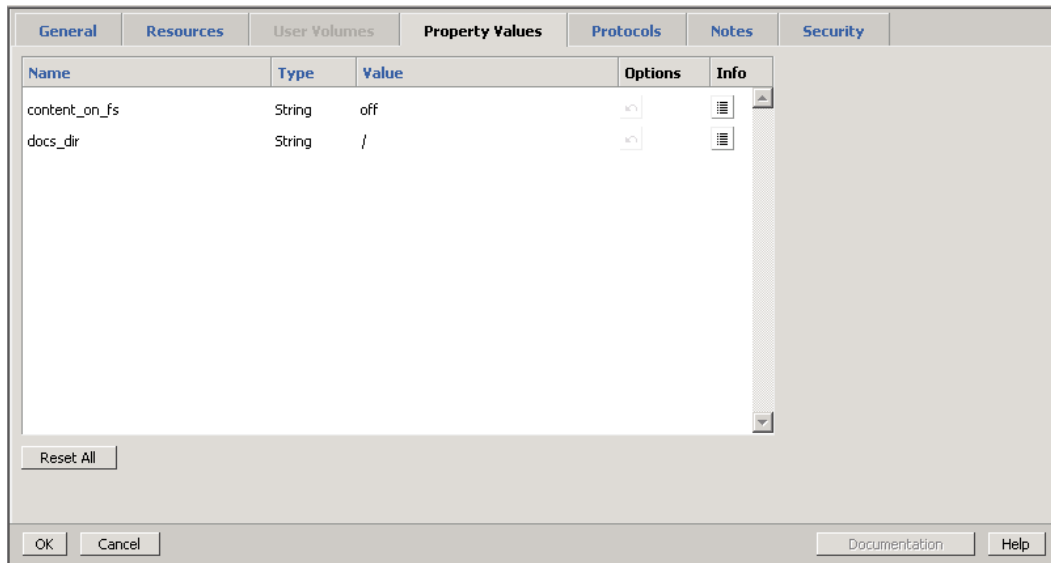


5. Click the **Save** button in the toolbar of the Application Editor.

We are now ready to configure the application. To do this:

1. Click the **Application** menu in the toolbar of the Application Editor, and select **Configure** in the context menu.
2. Click the **Property Values** tab.
3. Leave the default values for this example for the two properties and click **OK** when done.

Note: In this configuration, AppLogic will define D:\ as the root directory for the content, since content_on_fs will be set to "off". Also, content will be stored on the Application User Volume.



4. Click the **Save** button in the toolbar of the Application Editor when done.

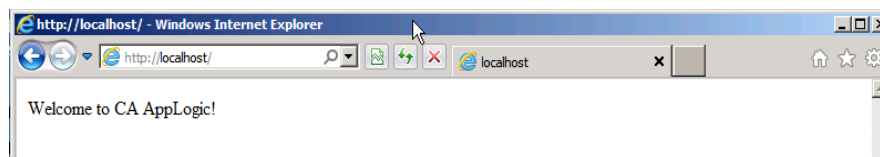
Step 4: Testing the Application

We can now verify that the application is working by creating a simple web page on the content volume and confirming that IIS can serve the web page. To do this, perform the following steps:

Note: For purposes of this exercise, we did not add an IN and NET gateway, therefore, the only access to the application is by logging in through the graphical console. To add external access into this web server you would need to add an IN gateway. To enable the IIS server or the MS SQL Server to communicate to an external network you would need to add a NET gateway appliance. In both cases, the IN and the NET would be placed at the same level as the assembly.

1. Start the application
2. Create a simple **default.htm** (with some text in it) file and copy it to the **docs_dir** directory (**D:** by default)
3. Launch **http://localhost** on the IIS server
4. Confirm that the default.htm page is displayed.

For example:



If you add an IN gateway appliance to the application you can test the assembly by launching the page from outside CA AppLogic and verifying that default web page is presented.

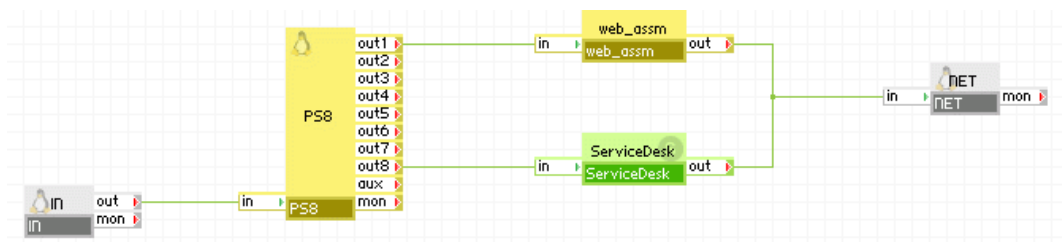
Step 5: Save the Assembly

Before we save the assembly within a catalog to transform the assembly into a class, we must first transform any singletons into appliances by saving them to the target catalog. Another good practice is to launch a Grid Shell and execute the **app clean** command.

Once this is done, save the assembly to a catalog by dragging and dropping the assembly onto the desired catalog category(**system_ms** for example).

Using the Assembly

You can now follow these steps to create assemblies for other applications – such as CA Service Desk Manager. These assemblies can then be captured in a single application to provide a one step solution to your clients' data center requirements.



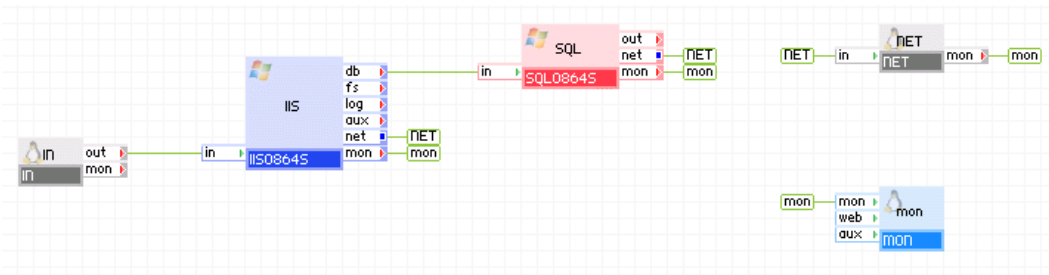
Create an Application with an External SQL Server

Depending on your particular environment's requirements, it may not be possible, or preferable, to include a Microsoft SQL Server within a CA AppLogic application. For example, if your site requirements dictate that all databases be located behind a firewall or on a dedicated server. In this section we will demonstrate how to modify the IIS Application we created earlier to accommodate those requirements – by creating an application that uses a SQL server that is external to the CA AppLogic application.

Before we do this, it is good practice to make a copy of the original IIS application so that it will still be available after changes are made to the copy. To do this, perform the following steps:

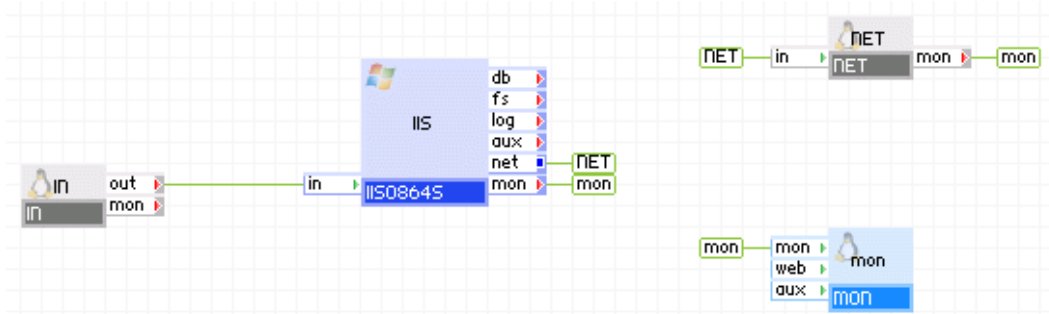
1. Locate the IIS application within the CA AppLogic GUI and verify that it is stopped.
If the application is running, right-click on it and select **Stop** in the context menu.
2. Right click the application and select **Copy** in the context menu.
3. Wait for the copy process to complete.
4. Click the new application to open the Application Editor.

For example:



5. Click the SQL0864S appliance, and press the **Delete** key.
6. Verify that there is a connection between the **net** terminal of the IIS0864S appliance and the **in** terminal of the NET appliance.
7. Verify there is a connection between the **mon** terminal of the NET appliance and the **mon** terminal of the MON appliance.
8. Right-click the **NET** appliance, and click the **Property Values** menu item in the context menu.
9. Verify that appropriate values have been specified for the **ip_addr**, **netmask**, **gateway**, and the **dns1** properties.
10. Click **OK** to save your changes in the Instance Settings window.
11. Click the **Save** button in the toolbar of the Application Editor.

Your application should now look similar to the following:



12. Right-click the application within the CA AppLogic GUI and click **Start** in the context menu.
13. Click the application within the CA AppLogic GUI to open the Application Editor.
14. Click the IIS0864S appliance, and click the **Login (graphic)** button to open the graphical console.
15. Login to Windows using the appropriate credentials.
16. Confirm you can ping the hostname of the external Microsoft SQL Server.
17. Open the **ODBC Data Source** application by executing **Start > Administrative Tools > Data Sources (ODBC)**.
18. Click the **System DSN** tab.
19. Click the **Add...** button.

20. Select **SQL Server** for the driver and click the Finish button.
21. Provide the following information on the first page of the wizard, and click **Next**:
 - Name: <Any name is fine>
 - Server: <external_SQL_Server_Name>

Note: It may be necessary to provide the fully qualified domain name.
22. Select the option *With SQL Server authentication using a login ID and password entered by the user*, then provide the credentials for the **sa** database user, and click **Next**.
23. Check the *Change the default database to:* checkbox, and select **master** from the drop-down list. Click **Next**.
24. Click the **Finish** button.
25. Click the **Test Data Source...** button and verify the message “**TESTS COMPLETED SUCCESSFULLY!**” is displayed.
26. Click the **OK** button.
27. Click the **OK** button in the **ODBC Microsoft SQL Server Setup** dialog.
28. Verify the System DSN that was just created is selected in the **ODBC Data Source...** dialog and click the **Remove** button to delete it.
29. When prompted to confirm to delete it, click the **Yes** button.
30. Click the **OK** button in the **ODBC Data Source...** dialog to close the ODBC Data Source Application.
31. The IIS0864S is now ready to connect to an external MS SQL Server.

Maximizing Appliance Re-usability

To fully leverage the agility that CA AppLogic provides to application developers you need to understand the boot process an appliance follows at startup as well as how specific boundary configuration items are used during this time. Understanding these concepts will enable you to go beyond simply installing the product set in an appliance or appliances to producing standalone building blocks that can support re-usability of base appliances while facilitating portability and instantiation of applications.

CA AppLogic provides boot time control over your appliances. For Linux/UNIX environments, this is done by leveraging the native shell environment to execute services and scripts provided by the CA AppLogic Appliance Kit (APK), specifically the `/etc/init.d/applogic_init` service, which will run the `/appliance/appliance.sh` script.

A similar process is followed for Windows based appliances, utilizing the appliance's Cygwin environment which is delivered through the VDS_Windows-X.X.X.X.msi (installed when you transformed the singleton object into a catalog class item). In this case, the Windows operating system is provided with the following two Windows services to control the boot level appliance execution:

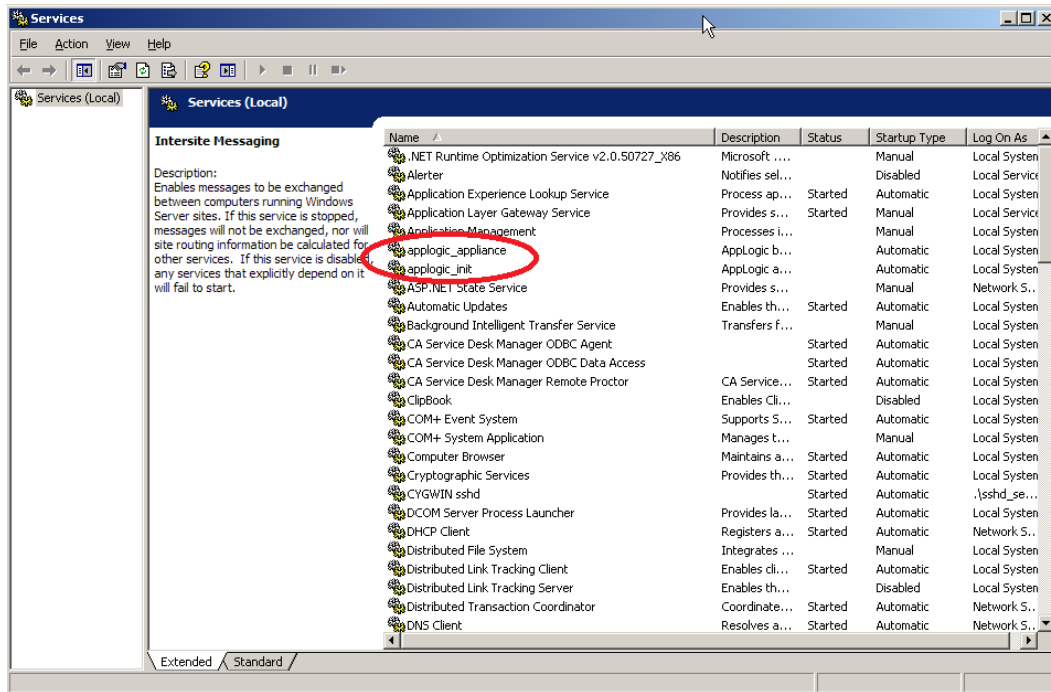
- **applogic_init**

Configures the appliance boot environment and receives configuration information through DHCP request to the grid controller.

- **applogic_appliance**

Runs and executes the scripts and configuration tasks.

Here you can see both services in the Windows appliance services list.



You can leverage this behavior by creating an environmental variable and assigning an initial value for each property defined by the appliance boundary upon startup. These properties can then be propagated into one or more of the configuration files contained on the boot volume. By extracting the configuration aspects of each appliance to the appliance level boundary definition and, ultimately, to the application level boundary, this enables you to use properties as the primary mechanism for configuring appliances.

Appendix A: Using the Microsoft Product Key Injection Utility Licensing Scripts

The Microsoft Product Key Injection Utility was designed to address the challenge of adding a valid product license key to CA AppLogic wrapped components which depend on the Microsoft operating systems and applications. The utility does this by embedding licensing scripts into base classes. These scripts update and activate the Microsoft products based on product keys that are set as component properties.

IMPORTANT: THE INFORMATION PROVIDED IN THIS SECTION AND IN THE ASSOCIATED UTILITY AND SCRIPTS SHOULD NOT IN ANY WAY BE INTERPRETED TO SUPERSEDE REQUIREMENTS SPECIFIED BY PRODUCT MANAGEMENT OR BY LEGAL. YOU ARE ADVISED TO VERIFY COMPLIANCE WITH CURRENT REQUIREMENTS BEFORE IMPLEMENTING THIS UTILITY.

Supported Microsoft Products

Product key injection scripts currently available:

- setProdKeyWin2008r2.ps1 for Microsoft Windows Server 2008 R2
- setProdKeySQL2008r2.ps1 for Microsoft SQL Server 2008 R2

Before You Begin

To complete the procedure successfully you should...

- Be familiar with CA AppLogic functions and terminology
- Have some experience using the BASH and AppLogic shell commands
- Be familiar with the *NIX “vi” editor
- Be familiar with PowerShell scripting
- Have an understanding of Microsoft product installation and licensing
- Be familiar with Microsoft group policy
- Be familiar with the Microsoft SLMGR utility

For additional details contact your grid administrator and/or consult the CA AppLogic documentation and support site.

Microsoft Product Installation

Detailed instructions for feature selection and complete installation procedures for Microsoft Windows and Microsoft SQL Server are beyond the scope of this document. Only steps necessary to support key injection are covered.

Microsoft Windows Server 2008 R2

Microsoft Windows Server 2008 R2 requires a product key to complete the installation. When installation, configuration and customization are complete, the key must be removed using the Microsoft licensing utility “slmgr.vbs”. The command syntax to remove the product key is as follows:

```
%windir%\system32\slmgr.vbs -upk
```

By default Windows will attempt to auto-activate in three days. To disable auto-activation do the following:

1. Navigate to the following key in the registry:

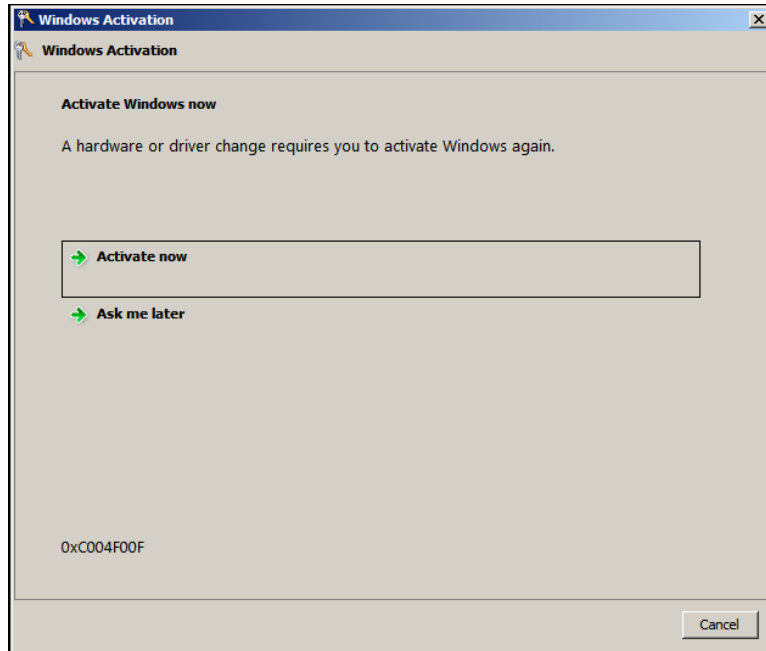
```
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\SoftwareProtectionPlatform\Activation
```

2. Create the DWORD value named “Manual” if it does not exist and set the value data to “1”.

When the product key is removed the following “This copy of Windows is not genuine” warning will be displayed in the lower right-hand corner of the desktop:



After reboot and login the following panel will be displayed:

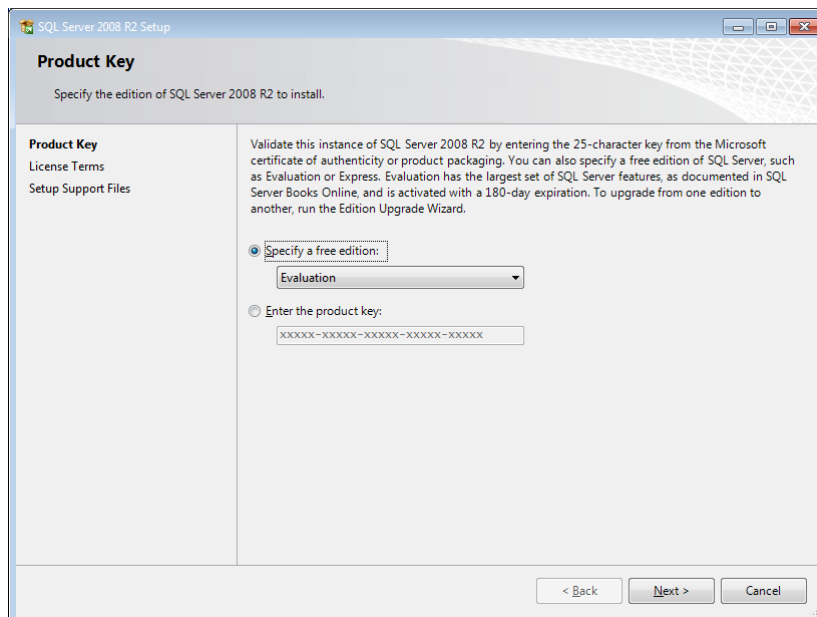


Choose “Ask me later” or click the **Cancel** button to continue without activating.

The desktop should be black (by default, a blue desktop indicates that Windows has been activated) and should display a “This copy of Windows is not genuine” warning message in the lower right-hand corner.

Microsoft SQL Server 2008 R2

Microsoft does not support a method to remove the product key for SQL Server 2008 R2 post-installation. Therefore, during the installation you will need to select “Specify a free edition” and “Evaluation” on the “Product Key” panel:

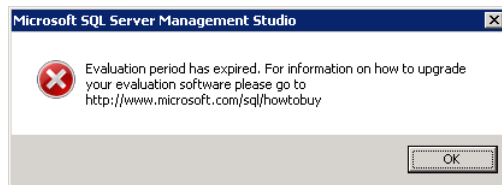


To verify Microsoft SQL Server 2008 R2 has been installed without a key for evaluation check the following registry key:

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL Server\100\Tools\Setup

The value for “Edition” should read “...Evaluation Edition”. If a product key was entered during installation, you will need to uninstall and re-install.

Important: The Microsoft SQL Server 2008 R2 evaluation period expires 180 days from the date installed. Any attempt to access Microsoft SQL Server Management Studio beyond the expiration date generates a warning message:



Microsoft does not support a method to re-arm or reset the expiration date.

Example Script Commands for Licensing

Following are some example commands that can be called from your scripts to help licensing Microsoft Windows Server, and Microsoft SQL Server:

Microsoft Windows Server

slmgr.vbs is the primary utility to use when licensing and activating Windows. Some examples of this command are:

- **slmgr.vbs -ipk <new_product_key>**: Updates the product key
- **slmgr.vbs -ato**: Activates Windows
- **slmgr.vbs -skms <kms_server_port>**: Sets the kms server name and port

For more Information on the slmgr.vbs utility, refer to <http://technet.microsoft.com/en-us/library/ff793433.aspx>

Microsoft SQL Server 2008 R2

C:\Program Files\Microsoft SQL Server\100\Setup Bootstrap\SQLServer2008R2\setup.exe is the primary utility to use when licensing Microsoft SQL Server. An example of a command that will license Microsoft SQL Server is:

```
“C:\Program Files\Microsoft SQL Server\100\Setup Bootstrap\SQLServer2008R2\setup.exe”  
/Q /ACTION=editionupgrade /INSTANCENAME=MSSQLSERVER /PID=<license_key>  
/IACCEPTSQLSERVERLICENSETERMS /SkipRules=Engine_SqlEngineHealthCheck  
Engine_OnlySupportedFeaturesUsedCheckSystem
```

Where *<license_key>* is the actual license key for Microsoft SQL Server 2008 R2.

Implementation

There are two primary steps to the procedure:

- Step 1: Install License Injection Script
- Step 2: Modify group policy

Step 1: Install the License Key Injection Utility Script

To install the utility do the following:

1. Create the following directory structure under the /appliance directory on the boot volume



2. Copy the following files to this directory structure:
 - Copy the PowerShell common utilities script “ps_common.ps1” to the “C:\appliance\init\util” directory if it does not already exist.
 - For Microsoft Windows 2008 r2 operating system copy the “setProdKeyWin2008r2.ps1” script to the “C:\appliance\init\mspk” directory.
 - For Microsoft SQL Server 2008 R2 host systems, copy the “setProdKeySQL2008r2.ps1” script to the same directory (“C:\appliance\init\mspk”).

Note: Use standard Windows tools to copy the files to ensure line terminators are not impacted.

3. Open a Cygwin shell session and execute following commands to update the owner, group and permissions of all directories and files in the “/appliance/init” directory:

```
chown -R root /appliance/init
chgrp -R SYSTEM /appliance/init
chmod -R 775 /appliance/init
```

4. In a Cygwin shell session, use the “vi” editor to add the following lines to the end of the /etc/sysconfig/applogic_appliance file:

For **Windows 2008 r2** Operating systems:

```
touch /appliance/init/mspk/~setProdKeyWin2008r2
```

For **Microsoft SQL Server 2008 r2** systems:

```
touch /appliance/init/mspk/~setProdKeySQL2008r2
```

5. Verify the entries and save the file.

- Execute the following command from a Windows command shell:

```
powershell "&{Set-ExecutionPolicy RemoteSigned}"
```

- Execute the following command to verify that the environment was updated:

```
powershell "&{Get-ExecutionPolicy}"
```

The string "RemoteSigned" should be returned.

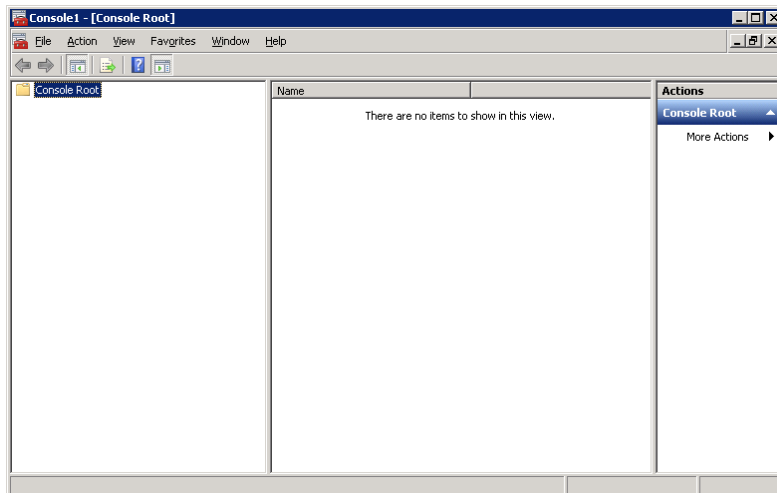
Step 2: Modify Group Policy

Scripts launched from CA AppLogic initialization scripts will run under the Cygwin 32-bit shell. To ensure that they will run as required in a 64-bit environment they should be launched as "Startup Scripts" using Windows group policy. To do this you need to do the following to modify the group policy for the script:

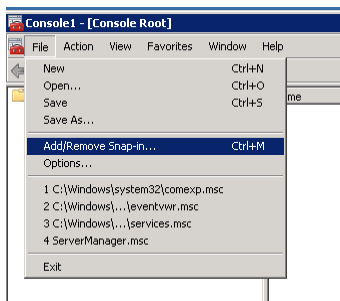
- Execute the following command:

```
Mmc.exe
```

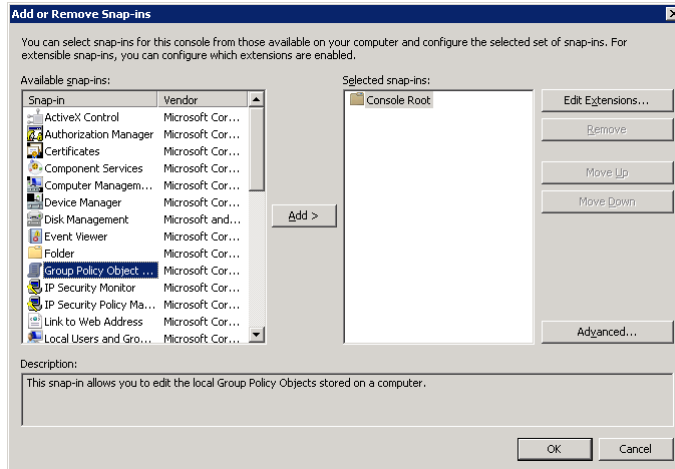
This opens a generic Windows management console.



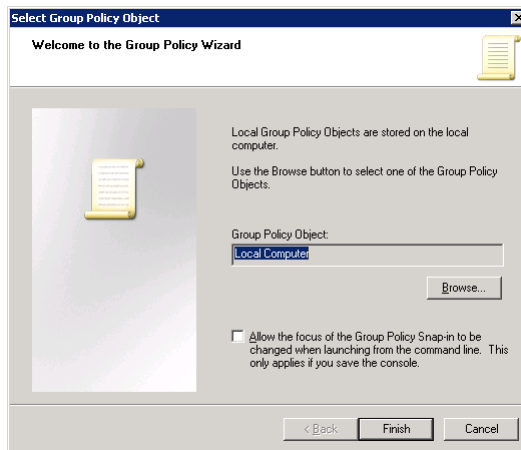
- Choose "Add/Remove Snap-in..." from the file menu or press <CTRL><M>.



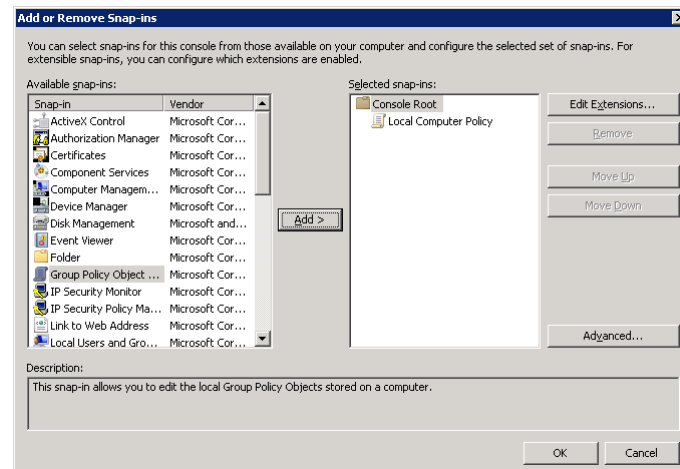
- Select "Group Policy Object" from the "Available snap-ins" list then click **Add**



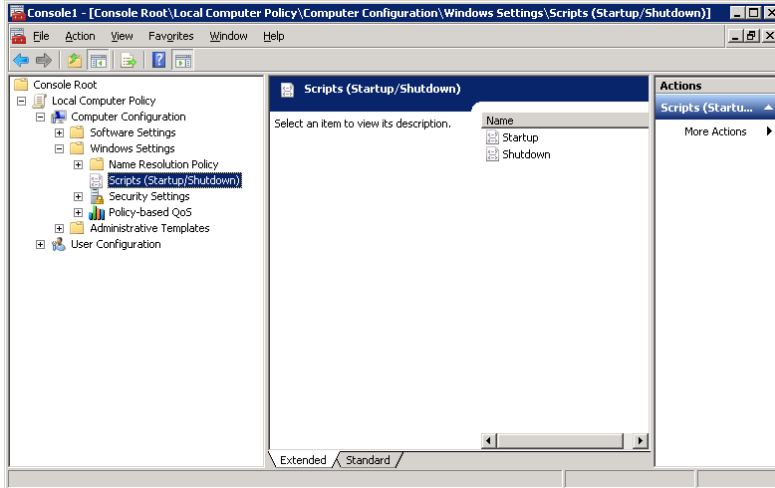
4. Click **Finish** on the “Select Group Policy Object” dialog.



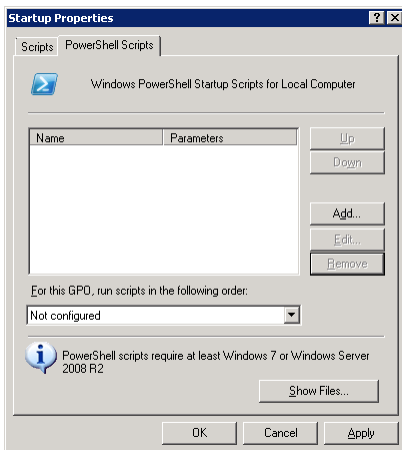
5. Verify that “Local Computer Policy” is displayed in the “Selected snap-ins” list:



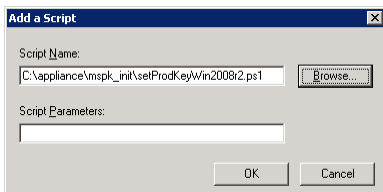
6. Click **OK**
7. Navigate to the “Scripts (Startup/Shutdown)” node (Console Root -> Local Computer Policy-> Computer Configuration -> Windows Settings) and double-click “Startup” in the right-hand pane.



8. Select the “PowerShell Scripts” tab then click **Add**

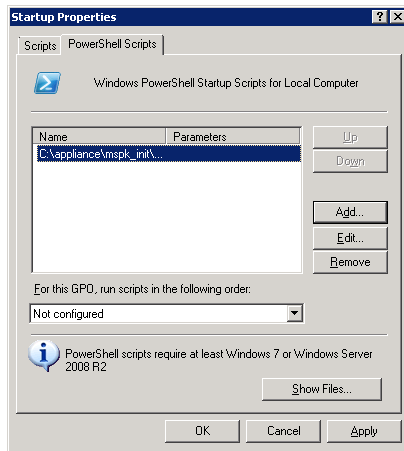


9. Provide the path and file name for the “setProdKeyWin2008r2.ps1” script (or use the **Browse** button).
10. To specify a default product key, enter a valid key in the “Script Parameters” text box. Otherwise, leave the “Script Parameters” empty and **OK**.



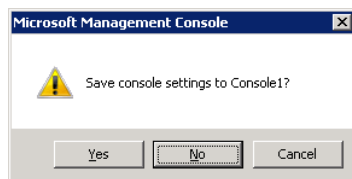
IMPORTANT: IF YOU SPECIFY A DEFAULT PRODUCT KEY THE SCRIPT WILL USE THAT KEY UNLESS A DIFFERENT KEY IS SPECIFIED WHEN THE COMPONENT IS STARTED. DEPENDING ON WHICH MICROSOFT PRODUCT IS BEING INSTALLED, IT MAY NOT BE POSSIBLE TO REMOVE THE LICENSE. FOR THIS REASON YOU SHOULD AVOID SPECIFYING DEFAULT LICENSE KEYS AND ALWAYS CONSULT FIRST WITH THE PARTY RESPONSIBLE FOR MICROSOFT LICENSING FOR THE COMPONENT.

11. If you need to use Microsoft SQL Server 2008 r2, add the “setProdKeySQL2008r2.ps1” script.
12. Verify the script(s) are listed then click **OK**.



The license injection scripts should now be able to run in the 64-bit environment each time Windows is started.

When closing the MMC console the following message may be generated:



Click **No** to exit without saving changes.

Script Execution

When the utility is correctly implemented the following will occur:

- The appropriate script(s) will be launched each time the Windows component is restarted (see [Modify Group Policy](#)).
- To ensure that the CA AppLogic component initialization is complete the code checks for the existence of specific semaphore files before continuing (see updates to the “/etc/sysconfig/applogic_appliance” file under [Step 2: Install the License Key Injection Utility](#)).
- License keys, which are set as component properties, are read from the “/var/run/applogic/appliance.conf” file and commands are executed to update and activate the Microsoft products as appropriate.
- Logs files are written to the “/appliance/init/logs” directory. Alerts and log messages are forwarded to the grid controller should any outcome require user intervention.

The exact behavior of each script will depend on how the properties are set.

setProdKeyWin2008r2.ps1 Script Properties

The behavior of the “setProdKeyWin2008r2.ps1” script is based on the values set for the following two properties:

- kms_server_port
- win2008r2_prod_key

kms_server_port

The “setProdKeyWin2008r2.ps1” script expects a value for an AppLogic component boundary, “kms_server_port” that is expressed in a format supported by the slmgr.vbs Microsoft licensing utility. That is:

- <KMS Name Resolvable to IPv4 Address>[:Port]
- <IPv4 KMS Address>[:Port]
- “None”
- Null

The property, if defined, should NOT be required so that it may be cleared for security reason once the component is updated successfully. If the property is not defined it is assumed to be null. Specifying the port number is optional and required only if the KMS server listening port is not set to the default.

Note: The current Microsoft licensing utility does not support IPv6 addresses.

The script will take the following actions depending on what values were specified:

Property Value	Behavior
Null	No action is taken No alert or log messages are sent to the CA AppLogic grid controller
None	Remove locally specified KMS Server entry. No alert or log messages are sent to the CA AppLogic grid controller.
specified	Attempt to update KMS server. If action is successful, no alert or log messages are sent to the CA AppLogic grid controller and the script execution continues. If the attempt fails, appropriate alert and log messages are sent to the CA AppLogic grid controller and script execution terminates.

WIN2008R2_PROD_KEY

Assuming a failure was not detected during the attempt to update the kms_server_port property (see [kms_server_port](#)), the “setProdKeyWin2008r2.ps1” script expects a value for the win2008r2_prod_key key that is expressed in the following format:

XXXXX-XXXXX-XXXXX-XXXXX-XXXXX

The property, if defined, should NOT be required so that it may be cleared for security reason once the component is updated successfully. If the property is not defined it is assumed to be null.

Based on the values specified for this property, the script will take the following actions:

Return Codes

The following return codes are generated by the “setProdKeyWin2008r2.ps1” script:

Return Code	Description
0	Script completed without exceptions
1	Script timed out waiting for system ready semaphore (see updates to the “/etc/sysconfig/appllogic_appliance” file under Install the License Key Injection Utility Script)
2	Product is not licensed and activated, product key was not specified
3	Product key specified is incorrectly formatted
4	Product key update failed
5	Product key activation failed
6	KMS server update failed

Detailed activity for troubleshooting and debugging is written to the “C:\appliance\init\logs\setProdKeyWin2008r2.log” file.

setProdKeySQL2008r2.ps1 Script Properties

The behavior of the “setProdKeySQL2008r2.ps1” script is based on the values set for the sql2008r2_prod_key property.

SQL2008R2_PROD_KEY

The “setProdKeySQL2008r2.ps1” script expects a value for the sql2008r2_prod_key that is expressed in the following format:

XXXXX-XXXXX-XXXXX-XXXXX-XXXXX

The property, if defined, should NOT be required so that it may be cleared for security reason once the component is updated successfully. If the property is not defined it is assumed to be null.

Based on the values specified for this property, the script will take the following actions:

Product Licensed	Property Value	Default Value	Behavior
No	Null	Null	No action is taken. Appropriate alert and log messages are sent to the CA AppLogic grid controller requesting action be taken to properly license the product.
No	Null	Specified	Attempt to update license using default value specified. If successful, no alert or log messages are sent to the CA AppLogic grid controller. If the attempt fails the appropriate alert and log messages are sent to the CA AppLogic grid controller.
No	Specified	Null	Attempt to update license using property value specified. If the attempt to update the product key is successful an appropriate log message is sent to the CA AppLogic grid controller log along with an alert requesting that the property value be cleared for security. If the attempt fails appropriate alert and log messages are sent to the CA AppLogic grid controller.
No	Specified	Specified	Default is ignored. Attempt to update license using the property value specified. If the attempt to update the product key and activate is successful an appropriate log message is sent to the CA AppLogic grid controller log along with an alert requesting that the property value be cleared for security. If the attempt fails appropriate alert and log messages are sent to the CA AppLogic grid controller.

Product Licensed	Property Value	Default Value	Behavior
Yes	Null	Null	<p>No action is taken.</p> <p>No alert or log messages are sent to the CA AppLogic grid controller.</p>
Yes	Null	Specified	<p>Default is ignored and no action is taken.</p> <p>No alert or log messages are sent to the CA AppLogic grid controller.</p>
Yes	Specified	Null	<p>If the product key specified matches the current product key no action is taken to update the product key.</p> <p>If the product key specified does not match the current product key an attempt will be made to update the product key using property value specified (supersede current license) and activate.</p> <p>If the product key matches the current product key or if the attempt to update the product key is successful an appropriate log message is sent to the CA AppLogic grid controller log along with an alert requesting that the property value be cleared for security.</p> <p>If the attempt to update the product key fails appropriate alert and log messages are sent to the CA AppLogic grid controller and the previous license remains in effect.</p> <p>If the attempt to activate fails appropriate alert and log messages are sent to the CA AppLogic grid controller, no product key is in effect and the product reverts to an evaluation edition.</p>
Yes	Specified	Specified	<p>Default is ignored.</p> <p>If the product key specified matches the current product key no action is taken to update the product key.</p> <p>If the product key specified does not match the current product key an attempt is made to update the license using the property value specified (supersede current license).</p>

Product Licensed	Property Value	Default Value	Behavior
			<p>If the product key matches the current product key or if the attempt to update the product key is successful an appropriate log message is sent to the CA AppLogic grid controller log along with an alert requesting that the property value be cleared for security</p> <p>If the attempt fails an appropriate alert and log messages are sent to the CA AppLogic grid controller and the previous license remains in effect</p>

See the [Troubleshooting](#) section for specific messages and recommended remediation.

Return Codes

The following return codes are generated by the “setProdKeyWin2008r2.ps1” script:

Return Code	Description
0	Script completed without exceptions
1	Script timed out waiting for system ready semaphore (see updates to the “/etc/sysconfig/applogic_appliance” file under Install the License Key Injection Utility Script)
2	Product is not licensed and activated, product key was not specified
3	Product key specified is incorrectly formatted
4	Product key update failed
5	Product key activation failed
6	KMS server update failed

Detailed activity for troubleshooting and debugging is written to the “C:\appliance\init\logs\setProdKeySQL2008rs.log” file.

License Removal

The utility does not support methods to remove keys once installed; however, there may be Microsoft utilities that can be used to revert back to an evaluation version for some products. In such cases, you will need to log on to the component directly to carry out the necessary actions.

Troubleshooting Tips

Microsoft Product Key Injection Utility scripts generate messages to help identify the root cause of potential failures. Depending on the message, severity and content the message may be written to any or all locations listed below:

Local Log – All messages will be written to a log file that will be created in the “C:\appliance\init\logs” named for the script generating the content.

AppLogic Logs – A specific subset of messages containing detailed information to identify root cause of potential failures that may require user intervention will also be forwarded to the AppLogic Grid Controller to be displayed on the “Logs” tab of the management console.

AppLogic Alert – A more limited subset of messages recommending specific user intervention will also be forward to the AppLogic Grid Controller as alerts that may be displayed on the “Dashboard” tab. See AppLogic documentation for additional actions that may be triggered by alerts.

The tables that follow detail:

- Messages
- Related cause and remediation actions
- Severity
- Location(s) where messages will be recorded

...for specific scripts.

Script "setProdKeySQK2008r2.ps1"

Message	Cause/Remediation	Severity	Local Log	AppLogic Logs	AppLogic Alert
Microsoft SQL Server 2008 R2 is licensed as <Current License Description> product key <Masked Product Key>	No product key property was specified. Product was licensed previously or licensed using a product key specified as a default. No action is required	INFO	Yes	No	No
Microsoft SQL Server 2008 R2 is licensed as <Current License Description> product key <Masked Product Key>. For security, please clear the corresponding application property value.	Product key property was specified. Product key was licensed previously using the key specified or updated to use the product key specified. Property key property should be cleared for security reasons.	ERROR	Yes	Yes	Yes
Microsoft SQL Server 2008 R2 licensing or activation error was detected. Contact the party for Microsoft product licensing in your organization. See Logs tab for more details.	Error attempting to update using the product key referenced in more detailed messages referenced more detailed messages sent to AppLogic Logs was detected or the product is not licensed and no product key was specified. See specific error message(s) will be sent to the AppLogic Logs and take appropriate action.	ERROR	Yes	Yes	Yes
Attempt to update Microsoft SQL 2008 R2 using product key <Masked Product Key> failed - <Microsoft Generated Details>	Attempt to update product key to product key referenced failed. See detailed message generated by Microsoft for root cause and take appropriate action.	ERROR	Yes	Yes	No

Message	Cause/Remediation	Severity	Local Log	AppLogic Logs	AppLogic Alert
Microsoft Windows SQL 2008 R2 appears to be incorrectly formatted – should be XXXXX-XXXXX-XXXXX-XXXXX	<p>Product key specified or default product does not match the format of a valid Microsoft product key.</p> <p>Check the product key and correct as appropriate.</p>	ERROR	Yes	Yes	No
Microsoft SQL Server 2008 R2 is currently not licensed and/or activated – product key was not specified	<p>Product is not licensed and no product key as was specified or set as a default.</p> <p>Product will continue to operate as an evaluation edition until a product key is specified or the evaluation period expires.</p>	ERROR	Yes	Yes	No
Microsoft Windows Server 2008 R2 license check waiting for system ready semaphore	<p>After launch, script waits 10 minutes (default) for a semaphore file to be created indicating the system is fully initialized (see Step 1: Install the License Key Injection Utility Script).</p> <p>Message indicates the script was launched and began to wait for creation of the semaphore file.</p> <p>No action is required.</p>	INFO	Yes	No	No
Waited <Time> milliseconds for system ready semaphore – starting license check process	<p>After launch, script waits 10 minutes (default) for a semaphore file to be created indicating the system is fully initialized (see Step 1: Install the License Key Injection Utility Script).</p>	INFO	Yes	No	No

Message	Cause/Remediation	Severity	Local Log	AppLogic Logs	AppLogic Alert
	<p>Message indicates the semaphore file was created within the allowed time period and the license check/update process was allowed to proceed.</p> <p>No action is required.</p>				
<p>Microsoft Windows Server 2008 R2 license check waited <Time> milliseconds for system ready semaphore – timed out</p>	<p>After launch, script waits 10 minutes (default) for a semaphore file to be created indicating the system is fully initialized.</p> <p>Verify the process described in Step 1: Install the License Key Injection Utility Script were completed correctly particularly items related to the updates to “/etc/sysconfig/applogic_appliance” file as well as changes to owner, group and permissions for the “/appliance/init” directory structure and content.</p>	ERROR	Yes	Yes	No

Script “setProdKeyWin2008r2.ps1”

Message	Cause/Remediation	Severity	Local Log	AppLogic Logs	AppLogic Alert
<p>Microsoft Windows Server 2008 R2 is licensed as <Current License Description> product key <Masked Product Key></p>	<p>No product key property was specified. Product was licensed previously or licensed using a product key specified as a default.</p> <p>No action is required</p>	INFO	Yes	No	No

Message	Cause/Remediation	Severity	Local Log	AppLogic Logs	AppLogic Alert
Microsoft Windows Server 2008 R2 is licensed as <Current License Description> product key <Masked Product Key>. For security, please clear the corresponding application property value.	<p>Product key property was specified. Product key was licensed previously using the key specified or updated to use the product key specified.</p> <p>Property key property should be cleared for security reasons.</p>	ERROR	Yes	Yes	Yes
Microsoft Windows Server 2008 R2 licensing or activation error was detected. Contact the party for Microsoft product licensing in your organization. See Logs tab for more details.	<p>Error attempting to update using the product key referenced in more detailed messages referenced more detailed messages sent to AppLogic Logs was detected or the product is not licensed and no product key was specified.</p> <p>See specific error message(s) will be sent to the AppLogic Logs and take appropriate action.</p>	ERROR	Yes	Yes	Yes
Attempt to update Microsoft Windows 2008 R2 using product key <Masked Product Key> failed - <Microsoft Generated Details>	<p>Attempt to update product key to product key referenced failed.</p> <p>See detailed message generated by Microsoft for root cause and take appropriate action.</p>	ERROR	Yes	Yes	No
Attempt to activate Microsoft Windows 2008 R2 using product key <Masked Product Key> failed - <Microsoft Generated Details>	<p>Attempt to product using product key referenced failed.</p> <p>See detailed message generated by Microsoft for root cause and take appropriate action.</p>	ERROR	Yes	Yes	No
Microsoft Windows	Product key specified or default product	ERROR	Yes	Yes	No

Message	Cause/Remediation	Severity	Local Log	AppLogic Logs	AppLogic Alert
Server 2008 R2 appears to be incorrectly formatted – should be XXXXX-XXXXX-XXXXX-XXXXX	<p>does not match the format of a valid Microsoft product key.</p> <p>Check the product key and correct as appropriate.</p>				
Microsoft Windows Server 2008 R2 is currently not licensed and/or activated – product key was not specified	<p>Product is not licensed and no product key as was specified or set as a default.</p> <p>Product will continue to operate as an evaluation edition until a product key is specified or the evaluation period expires.</p>	ERROR	Yes	Yes	No
Attempt to update KMS server failed - <Microsoft Generated Details>	<p>KMS server property other than “None” was specified. Attempt to use the Microsoft “slmgr –skms <KMS Server>” to update the system failed.</p> <p>Check the detailed message generated by the Microsoft to identify the root cause and take appropriate action.</p>	ERROR	Yes	Yes	No
<Microsoft Generated KMS Server Update Succeeded Message>	<p>KMS server property other than “None” was specified. Attempt to use the Microsoft “slmgr –skms <KMS Server>” to update the system succeeded.</p> <p>No action is required.</p>	INFO	Yes	No	No
<Microsoft Generated KMS Server Remove Succeeded Message>	<p>KMS server property “None” was specified. Attempt to use the Microsoft “slmgr –ckms” to update the system succeeded.</p> <p>No action is required.</p>	INFO	Yes	No	No
Microsoft Windows Server 2008 R2 license check waiting	<p>After launch, script waits 10 minutes (default) for a semaphore file to be created indicating the system is fully</p>	INFO	Yes	No	No

Message	Cause/Remediation	Severity	Local Log	AppLogic Logs	AppLogic Alert
for system ready semaphore	<p>initialized (see Step 1: Install the License Key Injection Utility Script).</p> <p>Message indicates the script was launched and began to wait for creation of the semaphore file.</p> <p>No action is required.</p>				
Waited <Time> milliseconds for system ready semaphore – starting license check process	<p>After launch, script waits 10 minutes (default) for a semaphore file to be created indicating the system is fully initialized (see Step 1: Install the License Key Injection Utility Script).</p> <p>Message indicates the semaphore file was created within the allowed time period and the license check/update process was allowed to proceed.</p> <p>No action is required.</p>	INFO	Yes	No	No
Microsoft Windows Server 2008 R2 license check waited <Time> milliseconds for system ready semaphore – timed out	<p>After launch, script waits 10 minutes (default) for a semaphore file to be created indicating the system is fully initialized.</p> <p>Verify the process described in Step 1: Install the License Key Injection Utility Script were completed correctly particularly items related to the updates to “/etc/sysconfig/applogic_appliance” file as well as changes to owner, group and permissions for the “/appliance/init” directory structure and content.</p>	ERROR	Yes	Yes	No

Appendix B: Microsoft SQL Server 2008 R2 64-bit Configuration File

This appendix contains the contents of the SQLSERVER2008 Configuration File.

Configuration File Contents

```
;SQLSERVER2008 Configuration File
```

```
[SQLSERVER2008]
```

```
; Specify the Instance ID for the SQL Server features you have specified. SQL Server directory structure, registry structure, and service names will reflect the instance ID of the SQL Server instance.
```

```
INSTANCEID="MSSQLSERVER"
```

```
; Specifies a Setup work flow, like INSTALL, UNINSTALL, or UPGRADE. This is a required parameter.
```

```
ACTION="Install"
```

```
; Specifies features to install, uninstall, or upgrade. The list of top-level features include SQL, AS, RS, IS, and Tools. The SQL feature will install the database engine, replication, and full-text. The Tools feature will install Management Tools, Books online, Business Intelligence Development Studio, and other shared components.
```

```
FEATURES=SQLENGINE,REPLICATION,FULLTEXT,AS,RS,BIDS,CONN,IS,BC,SDK,BOL,SSMS,ADV_SSMS,SNAC_SDK,OCS
```

```
; Displays the command line parameters usage
```

```
HELP="False"
```

```
; Specifies that the detailed Setup log should be piped to the console.
```

```
INDICATEPROGRESS="False"
```

```
; Setup will not display any user interface.
```

```
QUIET="False"
```

```
; Setup will display progress only without any user interaction.
```

```
QUIETSIMPLE="False"
```

```
; Specifies that Setup should install into WOW64. This command line argument is not supported on an IA64 or a 32-bit system.
```

```
X86="False"
```

```
; Detailed help for command line argument ENU has not been defined yet.
```

```
ENU="True"
```

```
; Parameter that controls the user interface behavior. Valid values are Normal for the full UI, and AutoAdvance for a simplified UI.
```

```
UIMODE="Normal"
```

```

; Specify if errors can be reported to Microsoft to improve future SQL Server releases.
Specify 1 or True to enable and 0 or False to disable this feature.
ERRORREPORTING="False"
; Specify the root installation directory for native shared components.
INSTALLSHAREDDIR="C:\Program Files\Microsoft SQL Server"

; Specify the root installation directory for the WOW64 shared components.
INSTALLSHAREDWOWDIR="C:\Program Files (x86)\Microsoft SQL Server"

; Specify the installation directory.
INSTANCEDIR="D:\Microsoft SQL Server"

; Specify that SQL Server feature usage data can be collected and sent to Microsoft.
Specify 1 or True to enable and 0 or False to disable this feature.
SQMREPORTING="False"

; Specify a default or named instance. MSSQLSERVER is the default instance for non-
Express editions and SQLExpress for Express editions. This parameter is required when
installing the SQL Server Database Engine (SQL), Analysis Services (AS), or Reporting
Services (RS).
INSTANCENAME="MSSQLSERVER"

; Agent account name
AGTSVCACCOUNT="NT AUTHORITY\SYSTEM"

; Auto-start service after installation.
AGTSVCSTARTUPTYPE="Manual"

; Startup type for Integration Services.
ISSVCSTARTUPTYPE="Automatic"

; Account for Integration Services: Domain\User or system account.
ISSVCACCOUNT="NT AUTHORITY\NetworkService"

; The name of the account that the Analysis Services service runs under.
ASSVCACCOUNT="NT AUTHORITY\SYSTEM"

; Controls the service startup type setting after the service has been created.
ASSVCSTARTUPTYPE="Automatic"

; The collation to be used by Analysis Services.
ASCOLLATION="Latin1_General_CI_AS"

; The location for the Analysis Services data files.
ASDATADIR="D:\Microsoft SQL Server\MSAS10.MSSQLSERVER\OLAP\Data"

; The location for the Analysis Services log files.
ASLOGDIR="D:\Microsoft SQL Server\MSAS10.MSSQLSERVER\OLAP\Log"

; The location for the Analysis Services backup files.
ASBACKUPDIR="D:\Microsoft SQL Server\MSAS10.MSSQLSERVER\OLAP\Backup"

```

```

; The location for the Analysis Services temporary files.
ASTEMPDIR="D:\Microsoft SQL Server\MSAS10.MSSQLSERVER\OLAP\Temp"

; The location for the Analysis Services configuration files.
ASCONFIGDIR="D:\Microsoft SQL Server\MSAS10.MSSQLSERVER\OLAP\Config"
; Specifies whether or not the MSOLAP provider is allowed to run in process.
ASPROVIDERMSOLAP="1"

; Specifies the list of administrator accounts that need to be provisioned.
ASSYSADMINACCOUNTS="WIN0864S\Administrator"

; A port number used to connect to the SharePoint Central Administration web application.
FARMADMINPORT="0"

; Startup type for the SQL Server service.
SQLSVCSTARTUPTYPE="Automatic"

; Level to enable FILESTREAM feature at (0, 1, 2 or 3).
FILESTREAMLEVEL="0"

; Set to "1" to enable RANU for SQL Server Express.
ENABLERANU="False"

; Specifies a windows collation or an SQL collation to use for the Database Engine.
SQLCOLLATION="SQL_Latin1_General_CP1_CI_AS"

; Account for SQL Server service: Domain\User or system account.
SQLSVCACCOUNT="NT AUTHORITY\SYSTEM"

; Windows account(s) to provision as SQL Server system administrators.
SQLSYSADMINACCOUNTS="WIN0864S\Administrator"

; The default is windows Authentication. Use "SQL" for Mixed Mode Authentication.
SECURITYMODE="SQL"

; Provision current user as a Database Engine system administrator for SQL Server 2008 R2 Express.
ADDCURRENTUSERASSQLADMIN="False"

; Specify 0 to disable or 1 to enable the TCP/IP protocol.
TCPENABLED="1"

; Specify 0 to disable or 1 to enable the Named Pipes protocol.
NPENABLED="0"

; Startup type for Browser Service.
BROWERSVCSTARTUPTYPE="Disabled"

; Specifies which account the report server NT service should execute under. When omitted or when the value is empty string, the default built-in account for the current operating system.
; The username part of RSSVACCOUNT is a maximum of 20 characters long and
; The domain part of RSSVACCOUNT is a maximum of 254 characters long.

```

```
RSSVCACCOUNT="NT AUTHORITY\SYSTEM"
```

```
; Specifies how the startup mode of the report server NT service. when  
; Manual - Service startup is manual mode (default).  
; Automatic - Service startup is automatic mode.  
; Disabled - Service is disabled
```

```
RSSVCSTARTUPTYPE="Automatic"
```

```
; Specifies which mode report server is installed in.  
; Default value: "FilesOnly"  
RSINSTALLMODE="FilesOnlyMode"
```

```
; Add description of input argument FTSVCACCOUNT  
FTSVCACCOUNT="NT AUTHORITY\LOCAL SYSTEM"
```

Glossary of Common Terms

Following are definitions of some of the terms and concepts used in this guide. For a more complete glossary of CA AppLogic terms consult the product documentation.

APK

The Appliance Kit (APK) is a package of tools and runtime for creating CA 3Tera AppLogic appliances out of OS installations. It contains scripts and utilities that help implement the appliance boundary, as well as a set of tools to assist appliance developers in preparing appliances.

Appliance

A copyable building block used to create AppLogic applications. The term appliance can be used to denote either the appliance class or an instance of it.

Application

A single system object that includes everything necessary to run a specific distributed application – application code, HTML pages, templates and scripts, databases and content, as well as the OS, middleware, file storage, load balancers, firewalls and all configuration information needed to reconstruct and run the application on an AppLogic grid. Each application also has a defined resource budget which specifies a minimum set of hardware resources (CPU, memory and bandwidth) required to run the application, and the maximum resource quota allowed for it.

Application Template

Application templates include all pieces required to establish a working instance of the solution – including the necessary database and input\output gateway appliances preconfigured for the specific architecture. They are designed to be provisioned to quickly stand up a working instance of the solution.

Assembly

Two or more appliances that are packaged together (sometimes called a “composite appliance”). An assembly represents a re-usable class encapsulating multiple devices pre-configured to communicate with each other and the “outside world” through the assembly boundary.

base appliance

A simple appliance that is designed to be used as the starting for creating additional appliances. Typically a base appliance represents a single operating system. For example, we created a Windows Server 2008 r2 base appliance in this guide and used it as the basis for creating additional appliance that required a Windows Server 2008 r2 operating system.

BFC

The Backbone Fabric Controller (BFC) is a graphical browser-based tool for creating, managing, and modifying CA 3Tera AppLogic grids. In CA 3Tera AppLogic version 2.9.9 and earlier, the administrator used a command line tool named the CA 3Tera AppLogic Distributor (ALDO) to create and manage their grids. The BFC replaces the previous manual static provisioning of servers to CA 3Tera AppLogic Grids with a container-based model that dynamically allocates servers to grids from a shared pool of resources under the control of the BFC.

Boot volume

A volume attached to the appliance which contains the operating system, configuration files and application software that runs an instance of the application for an appliance.

Boundary

Identifies the class name, input and output terminals, storage volumes, configuration values and defaults that comprise the definition of an appliance.

Catalogs

Set of disposable infrastructure appliances, such as gateways, firewalls, load balancers, web servers, application servers, database servers, file servers, and mail servers. The main assembly of an application ties them together into a logical structure capable of running the application. This includes all information required to configure each appliance and tie them together. When you create a new appliance you can add the class to the **local catalog** where it can be used by the application or add it to the **global** catalog, where it can be accessible to other applications and users.

Class

Re-usable device definition whose boundary attributes can be modified to adapt the device's behavior for a specific purpose. For example, after dragging a web server appliance class from the catalog onto the canvas, you can create a **singleton instance** of that web server class and modify the boundary properties to suit your particular environment. Those changes can even be saved as a new class.

Class definition

Identifies the class descriptor and one or more class volumes that comprise a particular class of virtual appliance.

Class instance

When you drag and drop an appliance class from the catalog onto the canvas this creates a single instance of that appliance which you can then modify and configure as needed.

To create an instance, AppLogic interprets the class descriptor and creates a virtual machine with one virtual network adapter for each terminal and a virtual block device for each volume. It then creates an instance of a virtual network interface for each of the adapters and binds it to the appropriate adapter. Next, the system creates a virtual volume instance for each volume specified in the descriptor by replicating the appropriate class volume and binds it to the corresponding block device.

Class volume

Contains all of the software required to boot and operate an instance of a class – includes the operating system, application server and anything else the application needs.

Component

A functional piece of a product's architecture. For example, a product may require a separate database server, which will be one component in the architecture.

Cygwin

Cygwin is an open source collection of tools that allows Unix or Linux applications to be compiled and run on a Windows operating system from within a Linux-like interface.

Fully Managed

Refers to an appliance which can be managed by CA AppLogic." For Windows appliances this is done by installing the Windows Server MSI, which installs the Windows Appliance Kit(APK) and it enables the following interactions between CA AppLogic and the appliance:

- The ability for the singleton/appliance to communicate with the grid controller by sending events if problems may occur.
- The ability to auto-configure the network interfaces of the singleton/appliance.
- The ability to obtain property values from the boundary of the singleton/appliance.

Gateway appliance

A specialized appliance which is used to manage the input and output of network traffic between applications and the outside world. Typical examples of gateway appliances are the IN and NET appliances.

Grid

The key component in grid computing. In the context of this document a "grid" refers to the combination of multiple computer resources that are combined and managed by AppLogic.

Grid Controller

Grid appliance that serves as the central point for managing the grid, creating, running and managing applications and monitoring operations.

Grid Nodes

The physical computers that comprise the grid on which AppLogic runs.

Grid Shell

Grid shell with the current application and or selected instance set as the current application and component.

Interior

Identifies the virtual machine and a boot volume that contains the operating system, configuration files and application software that runs an instance of the application for an appliance

PuTTY

A free telnet/SSH client that provides standard Windows type GUIs that can be used to perform SSH tasks. Although there are other tools which can be used to perform these tasks, the procedures provided in this document utilize the PuTTY suite – which includes PuTTY, PuTTY Pageant and PuTTYgen – to perform key installation tasks.

Singleton

An uncopyable appliance. In the AppLogic editor a singleton is indicated by an “S” symbol. Singletons are often used to edit or troubleshoot code.

SSH – Secure Shell

Network protocol that allows data to be exchanged using a secure channel between two networked devices.

Terminal\ Interface

Terminals are connection points for logical interactions between appliances – designed so that existing software packages inside virtual appliances can communicate through terminals without requiring modification. Looking from inside an appliance, the terminal is a host name visible only to that appliance instance. Terminals can be input or output. The terminal name of an input terminal can be used inside the appliance to set up a listening socket for accepting connections. The terminal name of an output terminal resolves to whatever appliance is connected to the output and can be used to establish connections to that appliance. When an output of one appliance is connected to an input of another appliance, AppLogic creates a virtual wire between their respective virtual network interfaces and assigns virtual IP addresses to both ends of the connection.