Oracle® Endeca Web Acquisition Toolkit

Administration Guide

Version 3.2.0 • January 2016

ORACLE

Copyright and disclaimer

Contents

Preface

This guide includes administration information for Kapow including:

- Runtime
- Tomcat Management Console

Related Documentation

In addition to the Administrator's Guide, the Kapow documentation set includes the followng documentation.

- Installation Guide
- Developer's Guide
- List other items here

Training

Kofax offers both classroom and computer-based training to help you make the most of your Kofax Kapow solution. Visit the Kofax website at www.kofax.com for details about the available training options and schedules.

Getting Help for Kofax Products

Kofax regularly updates the Kofax Support site with the latest information about Kofax products.

To access some resources, you must have a valid Support Agreement with an authorized Kofax Reseller/Partner or with Kofax directly.

Use the tools that Kofax provides for researching and identifying issues. For example, use the Kofax Support site to search for answers about messages, keywords, and product issues. To access the Kofax Support page, go to www.kofax.com/support.

The Kofax Support page provides:

- Product information and release news
  Click a product family, select a product, and select a version number.
- Downloadable product documentation
  Click a product family, select a product, and click **Documentation**.
- Access to product knowledge bases
  Click **Knowledge Base**.
- Access to the Kofax Customer Portal (for eligible customers)
  Click **Account Management** and log in.

To optimize your use of the portal, go to the Kofax Customer Portal login page and click the link to open the *Guide to the Kofax Support Portal*. This guide describes how to access the support site, what to do before contacting the support team, how to open a new case or view an open case, and what information to collect before opening a case.

- Access to support tools

  Click **Tools** and select the tool to use.
- Information about the support commitment for Kofax products

  Click **Support Details** and select **Kofax Support Commitment**.

Use these tools to find answers to questions that you have, to learn about new functionality, and to research possible solutions to current issues.

Chapter 1

Introduction

This guide is intended for system administrators who deploy Kapow in the enterprise environment.

Chapter 2

Runtime

Kapow offers a number of tools for executing the robots you have developed. The following sections describe these tools:

- RoboServer is a server application that allows remote clients to execute robots. It is configured using both the Management Console and the RoboServer Settings application (for advanced configuration, such as security and authentication).
- The Management Console allows you to schedule execution of Robots, view logs and extracted data. Also provides a dashboard for monitoring system health, and a centralized place where settings for clusters of RoboServers can be configured.

RoboServer

RoboServer runs robots created in Design Studio. Robots can be started in various ways; either scheduled to run at specific times by the Management Console, called via a REST web service, through the Java or .NET APIs or from a Kapplet.

Important The minimal Linux installation must include the following packages to be able to run Robots created with Default browser engine.

- `libX11.so.6`
- `libGL.so.1`
- `libXext.so.6`

Use `yum install` or `sudo apt-get` command to install necessary libraries on a Linux platform.

In order for RoboServer to be able to execute robots, it must be activated by a Management Console. A RoboServer is active when it belongs to a cluster in a Management Console with a valid license, and sufficient KCUs have been assigned to the cluster. The RoboServer also receives settings from the Management Console where they are configured on the clusters. Please refer to Management Console for more information on the administration of RoboServers and clusters.

Starting RoboServer

RoboServer can be started in several different ways:

- By clicking the RoboServer program icon (or the Start Management Console program icon which starts both the Management Console and RoboServer).
- By invoking it from the command line, which is described in detail below.
- By running it as a service. For more information about running RoboServer as a service, see Starting Servers Automatically.

To invoke RoboServer from the command line, open a Command Prompt window and type:

```
RoboServer
```

After you press Enter, the following help text appears in the command window before RoboServer terminates:

```
Detected the following plugins:
 ...

 Usage: RoboServer [-maxClippingSessions <num>] [-verbose] [-version] [-MC] [-port] [-
sslPort]
[-help] -service <service:params>

 Available services:
 ...
```

The dots indicate information that depends on your installation. If something different than the preceding is printed in the command window, please confirm that you have installed RoboServer as described in the Installation Guide.

RoboServer Parameters

Regardless of how you start RoboServer, it accepts the parameters in the following table.

Parameter	Description
`-c <num>|` `-maxClippingSessions <num>`	This parameter specifies the maximum number of clipping sessions that can exist on this RoboServer. This parameter is optional. The default value is 50. The minimum value is 0. Example: `-maxClippingSessions 20`
`-v |` `-verbose`	This optional parameter causes RoboServer to output status and runtime events.
`-V |` `-version`	This optional parameter causes RoboServer to output the version number, and then exit.
`-MC`	This optional parameter triggers the Management Console to be started as part of RoboServer. The Management Console runs on an embedded web server configured through the Settings application.
`-s <service-name:service-parameter> |` `-service <service-name:service-parameter>`	This parameter specifies a RQL or JMX service that RoboServer should start. This parameter must be specified at least once, and may be specified multiple times to start multiple services in the same RoboServer. The available services depend on your installation. Example: `-service socket:50000` Example: `-service jmx:50100`
`-p <port-number> |` `-port <port-number>`	This is shorthand for calling `-s socket:<port-number>` Example: `-port 50000`

Shutting Down RoboServer

RoboServer can be shut down using the command line tool ShutDownRoboServer. Run ShutDownRoboServer without arguments to see the various options for how to shut down the server, particularly how to handle any robots currently running on the server.

Production Configuration

RoboServer runs robots created with Design Studio. Robots can be started in various ways; either scheduled to run at specific times by the Management Console, called via a REST web service, through the Java or .NET APIs or from a Kapplet.

In order to get a stable and performing production environment, you may have to tweak some of the default RoboServer parameters. We will look at the following configuration options:

- Number of RoboServer instances
- Memory allocation
- Number of concurrent robots
- Automatic memory overload detection

RoboServer runs on Oracle's Java Virtual Machine (JVM), which in turn runs on top of an operating system (OS), which runs on top of your hardware. JVM's and OS's are patched, hardware architecture changes, and each new iteration aims to bring better performance. Although we can give some general guidelines about performance, the only way to make sure you have the optimal configuration is to test it.

As a general rule you get a little more performance by starting two instances of RoboServer. The JVM uses memory management known as garbage collection (GC). On most hardware only a single CPU core is active during GC, which leaves 75% of the CPU idle on a quad-core CPU. If you start two instances of RoboServer, one instance can still use the full CPU while the other in running GC.

The amount of concurrent robots a RoboServer can run depends on the amount of CPU available, and how fast you can get the data RoboServer needs to process. The number of concurrent robots is configured in the Management Console cluster settings. A robot running against a slow website will use a lot less CPU than a robot running against a website with a fast response time, and here is why. The amount of CPU used by a program can be described with the following formula

```
CPU (core)% = 1 - WaitTime/TotalTime
```

If a robot takes 20 seconds to execute, but 15 seconds are spent waiting for the website, it is only executing for 5 seconds, thus during the 20 seconds it is using an average of 25% (of a CPU core). The steps in a robot are executed in sequence, which means that a single executing robot will only be able to utilize one CPU core at a time. Most modern CPUs have multiple cores, so a robot that executes in 20 seconds, but waits for 15 seconds, will in fact only use about 6% of a quad-core CPU.

By default RoboServer is configured to maximally run 20 robots concurrently. The number of concurrent robots is configured in the Management Console cluster settings. If all your robots use 6% CPU, the CPU will be fully utilized when you are running 16-17 robots concurrently. If you start 33 of these 6% robots concurrently, you will overload RoboServer; because the amount of CPU available is constant, the result is that each robot will take twice as long to

finish. In the real world the CPU utilization of a robot may be anywhere between 5-95% of a CPU core, depending on robot logic and the website it interacts with. As a result it is hard to guess or calculate the correct value for the max concurrent robots, the only way to be sure you have the right value is to do a load test and monitor the RoboServer CPU utilization, as well as the robot runtime as load increases.

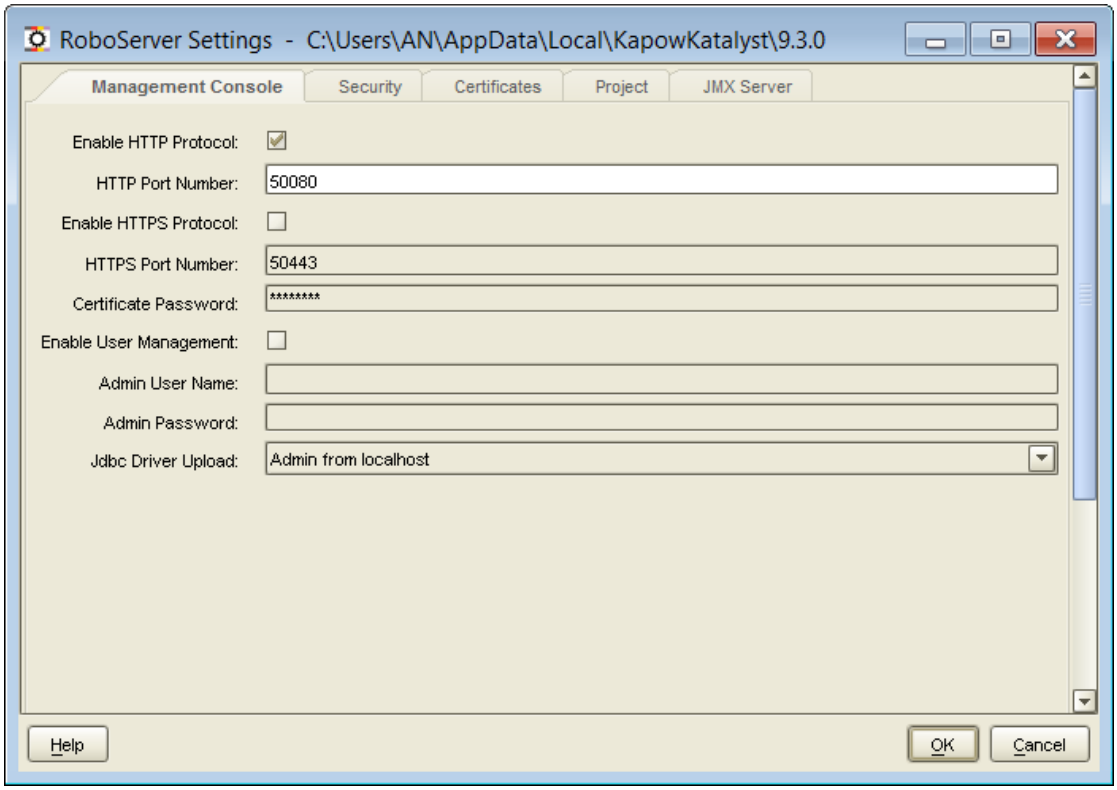
Another parameter that may affect the number of concurrent robots each RoboServer can handle is the amount of memory. The amount of memory used by robots can vary from a few megabytes (MB) to hundreds of MB. By default RoboServer is configured with 1024MB of memory, this is often not enough if you are actually executing 20 robots concurrently; check Changing the RAM Allocation to see how to control memory allocation. If you don't provide enough memory to RoboServer, you run the risk of crashing it with an out of memory error. The easiest way to ensure proper memory allocation is to monitor memory utilization during your load tests. If you allocate 2048 MB memory to a RoboServer, the JVM will not allocate all of it up front, but it will reserve it from the OS (this is why allocating more than 1200 MB frequently fails on 32bit Windows). Once the JVM starts to use the memory, it will not be given back to the OS. To find the optimal memory allocation you often have to run a series of load tests that push the CPU to 100%. After each test is complete, you check how much of the reserved memory was actually used by the JVM (the java.exe process). If all 1024MB (default) were used, increase (usually double) the memory and run the test again. At some point the JVM will not have used all of the reserved memory, and whatever it did use reflects the actual memory requirement and should be used to configure RoboServer.

Since RoboServer will crash if it runs out of memory, RoboServer tries to prevent this from occurring. Before RoboServer starts a new robot it will check the memory utilization. If it is above 80% it will queue the robot instead of starting it; this greatly reduces the risk of crashing RoboServer if the memory allocation is configured incorrectly. This mechanism is often referred to as the 80% memory threshold. The threshold value is configurable through the system property kapow.memoryThreshold=80.

RoboServer Configuration

You can configure RoboServer through the RoboServer Settings application. RoboServer Settings can be started from the Windows Start menu.

Settings Main Window



Using this application, you can configure the following:

- Management Console: Configuration of the embedded Management Console.
- Security: Security settings such as authentication and permissions.
- Certificates: The use of certificates.
- Project: The location of the default v project.
- JMX Server: The JMX server.

After changing any of the settings, click OK to store the new settings, and then restart any RoboServers that are running, to make the changes take effect.

Kapow contains several command-line tools to help you modify the settings in batch mode. For example, you can create several users with specified permissions. See Configuring RoboServer in Headless Mode.

If you need to change the maximum amount of RAM that RoboServer can use, see Changing the RAM Allocation.

Embedded Management Console Configuration

RoboServer contains an embedded web server which runs the Management Console. The web server is part of RoboServer, but is activated only when RoboServer is started with the -MC option enabled. By default, the web server will listen on port 50080, and thus the Management Console web interface is available on:

```
http:// host:50080/
```

Protocols and Ports

You can configure the web server to be accessible through HTTP and HTTPS on separate ports. If a protocol is enabled, a port number must be chosen; the defaults are port 50080 (HTTP) and port 50443 (HTTPS).

To enable HTTPS, a server certificate in JKS format must be stored in a file called tomcat.keystore in the `Certificates/Web` folder in the installation. If a certificate password other than the default (*changeit*) must be used, enter it in the Certificate Password field.

Enabling Administration Security

The Management Console can be accessed not only from the same computer (localhost), but also from others. One of the points of having a Management Console is that it coordinates execution of robots, and thus it typically must be accessible to many clients.

To mitigate the potential security risk of having access to the Management Console from other machines, you can enable an administrator password. Select the Use Admin Password option and enter the desired administrator user name and password. You must use these credentials both when you publish a robot to the Management Console from Design Studio and when you access the web interface from a browser.

You can also restrict who is allowed to upload JDBC driver to the embedded Management Console (see more about uploading JDBC drivers here). Possible choices are Not Allowed, where no one can upload JDBC drivers, "Admin from localhost," which means that the admin user can upload drivers when accessing the Management Console from the local machine; and finally, "Admin from any host," which means the admin user can always upload JDBC drivers.

Security

On the RoboServer settings Security tab, you specify general security restrictions, whether authentication is required for accessing the RoboServer, and audit logging preferences.

Restrictions

You can specify whether the RoboServer is allowed file system and command line access. By default, this is not allowed. If you enable it, however, robots running on RoboServer are allowed to access the file system and, using the Execute Command Line step, execute arbitrary commands on the machine running RoboServer.

Caution Enabling file system and command line access IS a security risk, and you should carefully consider whether it is necessary. If enabled, you should make sure the machine is not accessible from outside the local network, and/or you should require user authentication. Having a RoboServer with file system and command line access running on a machine accessible from the Internet and not requiring authentication, opens up the machine to the outside, and anyone can modify the file system according to the access rights of the user running RoboServer.

You can also disable accepting JDBC drivers from the Management Console. When activating RoboServers, the Management Console also sends settings to them. By default, this includes any JDBC drivers that have been uploaded to the Management Console. If a malicious user has gained administrator access to the Management Console, he could upload equally

malicious jar files which would then be sent to the RoboServers. If the admin Management Console user is only allowed to upload JDBC drivers from the localhost, the preceding would occur only if the attacker is in fact sitting in front of the machine running the Management Console, or has gained access to, for instance, a VPN (in which case you probably have bigger problems). So in general, it should not be necessary to disable accepting JDBC drivers. If you do, you can make JDBC drivers available to the RoboServer by manually putting them into the lib/jdbc directory of the installation folder as described here.

Requiring Authentication

To protect your RoboServer against unauthorized access, you can turn on authentication. This has effect on all RoboServers run from your Kapowinstallation, including a RoboServer started as a service or from a command line.

To turn on authentication, select the Require RoboServer Authentication check box. To run robots on a v with authentication turned on, you have to add users by clicking the add button. This will insert a new unnamed user. You can then fill out the information about the user including the username which will then be shown in the list of users.

A user is configured using the properties in the following table.

User Properties

Property	Description
Username	This is the username used by the user when accessing the v.
Password	This is the password used by the user when accessing the RoboServer.
Comments	Here you can write a comment about the user.
Start Robot	This enables the user to start robots on the RoboServer.
Stop Robot	This enables the user to stop robots on the RoboServer.
Shutdown RoboServer	This enables the user to shutdown the RoboServer from the Management Console.

To run robots on a RoboServer configured with authorization, the caller must provide proper credentials. In the Management Console, this is done in the settings. When running a robot via the Java API, credentials are provided as explained in Execution Parameters.

Configuring Audit Logging

To automatically have every HTTP and FTP request made by RoboServer logged, select the Log HTTP/FTP Traffic option. This will log HTTP and FTP requests using the Log4J logger specified by log4j.logger.kapow.auditlog. Log4J is configured by the log4j.properties file in the Configuration folder in the application data folder.

The audit log includes all requests to both web pages and all of its resources. Here's an excerpt from the log produced by loading the front page of Google:

```
02-29 13:51:21  INFO kapow.auditlog - google google.com http://google.com/ 301 0
```

```
 02-29 13:51:21  INFO kapow.auditlog - google www.google.com http://www.google.com/
 200 81688
 02-29 13:51:22  INFO kapow.auditlog - google www.google.com http://www.google.com/
extern_js/f/CgJkYRICZGsrMEUjtw.js 200 328960
 02-29 13:51:22  INFO kapow.auditlog - google www.google.com http://www.google.com/
extern_chrome/d9924a47b8c72e1a.js 200 43796
 02-29 13:51:23  INFO kapow.auditlog - google ssl.gstatic.com http://ssl.gstatic.com/
gb/js/sem_6501b4b3093bbedb61d2e.js 200 31642
```

The log contains the following information:

- Timestamp
- Log level (INFO)
- Logger (kapow.auditlog)
- Robot name
- Hostname
- Request URL
- HTTP/FTP response code
- Number of bytes loaded from the response body

Certificates

A key problem in establishing secure communications is to make sure that you're communicating with the right party. It is not sufficient just to request identity information from the other party - there must also be a way to verify this information before you can trust it. Certificates provide a solution to this, as they embody both a party's identity information (including that party's hostname and public key) and a signature from a trusted third party who vouches for the correctness of the identity information. A certificate should be trusted only:

- If the hostname stated within the certificate matches the hostname of the site that it comes from (otherwise it is a record of someone else's identity, which amounts to using false credentials).
- And if the certificate is signed by a third party that you trust.
- (Additionally, the certificate expiration date and the like should be checked, but we will not be concerned with these details here.)

We will not go into the technical details of the signing process, other than mentioning that it is based on public/private key cryptography. A signature for a certificate is the (fairly compact) result of a complex calculation that involves both the contents of the certificate and the signer's private key, and which cannot be reproduced without that same private key. The signature can be verified by doing another calculation that involves the contents of the certificate, the signature and the signer's public key. This calculation will tell whether the certificate matches the signature and thus is genuine. Note that the public key only enables you to verify a signature, not generate one. Thus the public key will not enable anyone to fake a certificate.

The signing party must never give the private key to anyone, but will distribute the public key as widely as possible. However, one issue still remains before you can trust a signature: You must be sure that you have a genuine copy of the signing party's public key. Public keys for well-known signing authorities like VeriSign are distributed with every browser and Java Runtime, and your trust in the signature (and thus the certificate) is in fact based on your trust in the way the browser or Java Runtime was installed.

It is possible to create your own signing authority by creating a public/private key pair and distributing the public key. This is done by embedding the public key in a certificate (a so-called self-signed certificate). Of course a party receiving such a certificate will not trust it based on its signature, but because he trusts you and the way that the certificate was communicated to him.

In order to make this scheme more flexible in actual implementation, it is possible to delegate the authority to sign. That is, the signature on a certificate may not actually be from a third party that you trust, but rather from yet another party who can display a certificate that you will trust. This may be extended to any number of levels, representing a chain of trust. To make verification practical, the signed certificate contains copies of all the certificates associated with the chain of trust (the last one being a self-signed root certificate). At least one of the certificates in this chain should be previously trusted by you.

Certificates are used in four different ways on RoboServer, corresponding to the four properties on the "Certificates" tab. Two of these have to do with how robots access web servers as part of their execution:

Verify HTTPS Certificates

A robot may need to verify the identity of a web server that it accesses (via HTTPS). Such a verification is routinely (and invisibly) done by ordinary browsers in order to detect phishing attacks. However, the verification often is not necessary when robots collect information, because the robots only access those well-known web sites that they have been written for. Thus the verification it is not enabled by default.

If enabled, verification is done in the same way a browser does it: The web server's certificate is checked based on an installed set of trusted HTTPS certificates similar to those you can configure in a browser.

HTTPS Client Certificates

This is almost the same as described above, but in the opposite direction. A robot may need to access a web server which wants to verify the identity of the client (robot) that accesses it. Presumably the web server contains confidential or commercial data that should be passed on only to clients with a proven identity. This identity is represented by a HTTPS client certificate.

Two other properties have to do with authentication in the communication between RoboServer and API clients that want robots to be executed on RoboServer. These properties apply only when clients connect to RoboServer via the secure socket based RQL protocol. The main purpose of the secure protocol is to encrypt communication, but with a little configuration it will also provide authentication, i.e., identity verification:

Verify API Client Certificates

RoboServer is able to verify the identity of any client that connects to it in order to execute robots. This verification is disabled by default.

If enabled, the mechanics of verification is the same as for HTTPS certificates even though the purpose is quite different. The connecting client's certificate is checked based on an installed set of trusted API client certificates.

API Server Certificate

RoboServer also has a server certificate that it will present to connecting clients. This certificate has a dual purpose: It makes the encryption side of SSL work (for this reason RoboServer comes with a default self-signed certificate), and it identifies this particular RoboServer to the clients.

The default API server certificate is the same for all RoboServers and thus is not any good for identification. If your clients need to verify the identity of the RoboServer they connect to, as described in SSL, you must create and install a unique API Server Certificate for each RoboServer.

Installing HTTPS Certificates

When a robot accesses a web site over HTTPS, it will verify the site's certificate (if the Verify HTTPS Certificates check box is selected). Verification is done based two sets of trusted certificates: the set of root certificates and an additional set of server certificates.

The root certificates are installed with Kapow just as root certificates are installed with your browser. They are found in the Certificates/Root folder in the application data folder.

Some HTTPS sites may use certificate authorities that are not included by default. In this case, you need to install the appropriate certificates for Kapow to load from these sites. Most often, these would be installed in the Certificates/Server folder in the application data folder.

To be precise, it does not matter - for the purpose of handling HTTPS sites - whether you add certificates to the set of root certificates or to the set of server certificates. However, please note that the root certificates have a broader scope, as they are also used when checking API client certificates.

To install a certificate, you need to obtain the certificate as a PKCS#7 certificate chain, as a Netscape certificate chain, or as a DER-encoded certificate. You install the certificate by copying it to either of the two folders mentioned above. The name of the file containing the certificate does not matter.

The following example explains how to install a server certificate for the website https://www.foo.com. The example is based on Internet Explorer.

Note You must install the certificates on all installations that need to load from the particular HTTPS sites.

1. Open https://www.foo.com in Internet Explorer.
2. Select **File** > **Properties** > **Certificates**.
   A Certificates window appears.
3. Click **Install Certificate**.
4. Click **Next** twice, click **Finish**, and exit the certificate window.
   Now you have installed the certificate in your browser. The next step is to export it to a file.
5. Click **Tools** > **Internet Options**.
   The Internet Options window appears.
6. On the Content tab, click the **Certificates** button.
7. Locate the installed certificate, which is typically on the Other People tab.
8. Select the certificate and click **Export**.
9. Click **Next** twice.
10. In the application data folder, save the certificate as foo.cer in the Certificates/Server subfolder.

11 Restart all your running RoboServers.

Installing HTTPS Client Certificates

When a robot accesses a HTTPS site, it may need to provide its own certificate to the web server. This is set up in the robot's Default Options or in the step-specific Options. One way to provide the certificate is to reference one of those that have been configured into the Kapowinstallation.

Note The HTTPS client certificates must be configured into all Kapow installations that need to run the robot.

1 On the RoboServer Settings Certificates tab, under the list of HTTPS client certificates, click the icon.

2 When prompted, select a certificate file, which must be in PKCS12 format.

3 Enter the password used to encrypt the certificate file.

4 Optionally change the unique ID assigned to the certificate by clicking the icon. The ID is used later in the robot to select the certificate.

Installing API Client Certificates

When API clients connect to RoboServer over SSL, RoboServer will verify the certificates presented to it (If the Verify API Client Certificates check box is selected). Verification means that RoboServer will reject connections from clients that fail verification, and is done based on two sets of trusted certificates: The set of root certificates and an additional set of API client certificates.

The root certificates are installed with Kapow just as root certificates are installed with your browser. They are found in the Certificates/Root folder in the application data folder. These are the same root certificates which are used for checking HTTPS certificates; however, root certificates probably will play a much smaller role when verifying API clients.

This is because in most cases, you will create your own self-signed API client certificates rather than use (expensive) certificates issued by official signing authorities. You should install your API client certificates in the Certificates/API/TrustedClients folder in the application data folder so that RoboServer will recognize them.

Technically speaking, it does not matter - for the purpose of verifying connecting API clients - whether you add API client certificates to the set of root certificates or to the set of API client certificates. However the guidelines given above will help you avoid problems caused by the fact that the root certificates are also (even mainly) used when checking HTTPS certificates.

You can generate a self-signed certificate for your API client with the Java keytool command as follows:

```
keytool -genkey -keystore client.p12 -alias client -keyalg RSA -storetype
"PKCS12"
```

You will be prompted for the following information: Name (domain), name of Organizational Unit, Organization, City, State, Country and password. Do not forget the password, there is no way to retrieve it if lost. This call of keytool will put the certificate into the keystore client.p12. You then must extract it into a separate file:

```
keytool -export -keystore client.p12 -alias client -storetype "PKCS12" -file
client.pub.cer
```

You will be prompted for the password used when the certificate was generated. The output file client.pub.cer is what should be copied into the Certificates/API/TrustedClients folder in the application data folder.

Installing API Server Certificate

For technical reasons, RoboServer must have a server certificate that it can present to API clients when they connect to it using SSL. During installation, a default self-signed certificate is installed. This certificate is invalid for identification purposes since it is the same for all RoboServers, and should be replaced if the API clients need to verify the identity of the RoboServer.

1. On the RoboServer Settings Certificates tab, click **Change**.
2. On the file selection window, select the certificate.
3. When prompted, enter the certificate password.
   The certificate is imported and the following properties appear: Issued To, Issued By, and Expires.
4. Use the following Java keytool command to generate a new self-signed certificate for RoboServer:
   ```
   keytool -genkey -keystore server.p12 -alias server -keyalg RSA -storetype
   "PKCS12"
   ```
5. When prompted, enter the following information:
   - Organizational Unit
   - Organization
   - City
   - State
   - Country
   - password

   Note Be sure to note the password, because it cannot be retrieved if lost.

   The certificate is saved in the keystore file server.p12, which can be imported.

Default RoboServer Project

You can set the location of the default RoboServer project folder in the on the RoboServer Settings Project tab. By default, the folder is set to the default robot project created in during the installation process. See the Design Studio User's Guide for more information on robot projects.

The RoboServer default project is used only by the API. When executing a robot from the API, any references it has to types, snippets or other resources are resolved by looking in the default project.

JMX Server Configuration

You can use the embedded JMX server to monitor the running RoboServer through tools such as JConsole. Enable JConsole by providing an argument on the RoboServer command line (see the RoboServer User's Guide for further information.)

Hiding Sensitive Robot Input

The Show Inputs option controls whether robot input parameters are shown in the management interface. This makes it possible to hide security sensitive information such as passwords.

JMX Server Access

By default, the JMX server can be accessed by all clients with access to the correct port on the server. By selecting the Use Password option, the selected user name and password are required when connecting.

Heartbeat Notifications

If an interval (in seconds) greater than 0 is specified, the JMX server sends out a heartbeat notification with the given interval, as long as the RoboServer is running and responding to queries.

RoboServer Configuration - Headless Mode

Kapow ships with several utilities to configure your RoboServer from a command line. The utilities are located in the bin subfolder of the Kapow installation folder. Note that the configuration files are user-dependent and stored in the user folder. For more information, see Important Folders in Kapow.

- ConfigureRS: Sets the JMX password in the RoboServer settings file (roboserver.settings).
- ConfigureMC: Sets Management Console administrator and certificate passwords in the mc.settings file.
- ConfigureRSUser: Adds and removes users and updates user credentials in the rsusers.xml file. Information in this file is used to authenticate API requests.

For help on usage, run utilities with an -h option.

To create a user user1 with Password1 password and all permissions type the next command:

```
ConfigureRSUser user1 Password1 -a
```

To enable authentication of API requests, you must open rsusers.xml and change the enabled attribute to true, as shown in the following example.

Sample rsusers.xml configuration file

```
<?xml version="1.0" encoding="UTF-8"?>

 <userConfiguration enabled="true">
   <users>
     <user username="user1"
 password_hash="d226d077a864683c45998a66fbb0157c681020c7628c31534b8718a1da00435505e4262e3f4dc305">
       <startRobot/>
       <stopRobot/>
       <shutdownRoboServer/>
     </user>
   </users>
```

```
 </userConfiguration>
```

Sample roboserver.settings configuration file

```
# Settings file for Management Console. Passwords should not be edited by hand, but
 using the 'ConfigureMC' command line utility.


 # Should the MC web-server start a HTTP listener. Values true/false
 mc_http = true

 # Configures the port of the http listener.
 mc_http_port = 50080

 # Should the MC web-server start a HTTPs listener. Values true/false
 mc_https = false

 # Configures the port of the HTTPS listener.
 mc_https_port = 50443

 # Password for the certificate used by the HTTPs listener. This should be created
 using the ConfigureMC command line tool.
 mc_https_cert_password = 3W2MTrL/b2k=

 # Enables MC internal user management, to support multi user scenarios. Values: true/
false
 mc_enable_usermanagement = true

 # The user-name of the MC super user.
 mc_admin_user =admin

 # The passwordHash of the MC super user. This is a salted SHA-256 hash.
 mc_admin_password
 =7800451255702ef8ae5f5fa0337833059d80b81d5af5872bdeafed230bab479896b6df4f63b25a24

 # Configures which hosts are allowed to upload JDBC jar files to MC. Values: NONE,
 LOCALHOST, ANY_HOST
 mc_allow_jdbc_upload = LOCALHOST
```

Sample mc.settings configuration file

```
# Settings file for Management Console. Passwords should not be edited by hand, but
 using the 'ConfigureMC' command line utility.


 # Should the MC web-server start a HTTP listener. Values true/false
 mc_http = true

 # Configures the port of the http listener.
 mc_http_port = 50080

 # Should the MC web-server start a HTTPs listener. Values true/false
 mc_https = false

 # Configures the port of the HTTPS listener.
 mc_https_port = 50443

 # Password for the certificate used by the HTTPs listener. This should be created
 using the ConfigureMC command line tool.
 mc_https_cert_password = 3W2MTrL/b2k=

 # Enables MC internal user management, to support multi user scenarios. Values: true/
false
 mc_enable_usermanagement = true

 # The user-name of the MC super user.
 mc_admin_user =admin

 # The passwordHash of the MC super user. This is a salted SHA-256 hash.
 mc_admin_password
 =7800451255702ef8ae5f5fa0337833059d80b81d5af5872bdeafed230bab479896b6df4f63b25a24

 # Configures which hosts are allowed to upload JDBC jar files to MC. Values: NONE,
 LOCALHOST, ANY_HOST
 mc_allow_jdbc_upload = LOCALHOST
```

Changing the RAM Allocation

As installed, each Kapow application is configured with a maximum amount of RAM that it may use. This amount usually is plenty for ordinary work, but if you run many robots in parallel on RoboServer, or if some robots use much RAM, it may be necessary to increase the allocation.

You can change the allocation for any of the applications by editing its .conf file, found in the bin subfolder of the installation. For RoboServer, the file to edit is named:

```
bin/RoboServer.conf
```

You need add a line with:

```
wrapper.java.maxmemory=1024
```

For example, to permit RoboServer to use up to 4GB of RAM, add this line:

```
wrapper.java.maxmemory=4096
```

Note An allocation this high is possible only on the 64-bit version of Kapow. Also, the .conf file can be edited only by the user who installed Kapow, such as the Windows administrator.

If necessary, the maximum RAM allocation for other Kapow applications can be changed in the same way.

Troubleshooting RoboServer Service Startup

If your service does not start, look for RoboServer messages in the Windows event log. Make sure you have installed the service with the wrapper.syslog.loglevel=INFO argument. For more information, see Kapow Initial Configuration in the Installation Guide.

Chapter 3

Tomcat Management Console

By default, Management Console is run as an embedded component inside RoboServer, which makes for easy installation. As an alternative, it can be deployed as a regular web application on a standalone Tomcat web server version 7.0.56 or later.

Important If your setup requires access to the Management Console outside of your corporate intranet, make sure SSL is set up to work with your Tomcat server.

The following table lists the differences in the feature set.

Management Console Features and Configuration

Feature	Embedded	Standalone J2SE Web Container
Authentication	Single-user defined in Settings. Users and Roles managed by Management Console Administrator.	Users and Roles managed by Management Console Administrator. Role based security through Active Directory or other LDAP provider. Single Sign-On using SiteMinder.
Management Console data store	Embedded Derby database	Container managed Data Source (supported platforms)

Note The derby JDBC driver is not distributed with the Enterprise Management Console. See Apache Derby Web site for Derby JDBC driver download information. We recommend using MySQL or other enterprise-class database with your Enterprise Management Console.

Instructions on configuring an embedded Management Console can be found in the Kapow online help. To start an embedded Management Console, see Starting the Management Console under the Management Console section.

Upgrade Notes

If you have been running a previous version of Management Console, this section will inform you how to upgrade to 9.4.0

Upgrading involves the following steps:

1. Create a backup of your data from the previous version of Management Console
2. Install Management Console
3. Update the Management Console configuration
4. Restore the backup

Create a Backup

Check the online documentation of your previous version of Management Console to learn how to create a backup. 9.2 , 9.1, 9.0, 8.2, 8.1, 8.0, 7.2, 7.1 and 7.0.

Users upgrading from version 9.2 can also consult our notes on upgrading and the Data Migration Tutorial section.

Install Management Console

Follow the next steps in this guide to learn how to install the Management Console. If you are upgrading from 8.1 or later you can reuse your old Configuration.xml, see next section.

Upgrade Management Console Configuration

If you are upgrading from version 8.0 or earlier, you will have to re-configure your new Management Console from scratch. Prior to version 8.1 some configuration was done in web.xml this is no longer the case. You may no longer copy or modify the web.xml.

Most configuration is done in Configuration.xml. This file may be copied from previous versions. Copy Configuration.xml from the previous version into WEB-INF/ overriding the version already there. Once you start Management Console, it will create a new Configuration.xml based on your old version. You must then copy the upgraded version into WEB-INF/ overriding your old configuration. Instructions should display when you access Management Console web interface.

As of version 8.2 , LDAP integration is no longer container-managed. If you used LDAP for authentication in a previous version, you should remove the Realm definition in the ManagementConsole.xml inside Tomcat's conf/Catalina/localhost/. LDAP integration is now configured in WEB-INF/login.xml, as described in LDAP Integration

Restore the Backup

Check the Management Console user's guide for details on how to restore a backup.

Tomcat Deployment

We will now detail how to install the Management Console on a stand alone J2SE web container. For this guide we have chosen Tomcat, and the distribution has been tested on Tomcat 7.0.56, 7.0.59, and 8.0.20. Your J2SE web container must be using the Java 8 runtime or later. For example, you can visit the Oracle Java SE Downloads site and download the latest Java 8 release.

Important If your setup requires access to the Management Console outside of your corporate intranet, make sure SSL is set up to work with your Tomcat server.

Configuring ManagementConsole.war

The Management Console application comes in the form of a Web Application Archive (WAR file) named ManagementConsole.war, which is located in the /WebApps folder in the Kapow installation folder.

The version of ManagementConsole.war that ships with Kapow is configured to run embedded inside RoboServer. Before you can deploy it as a standalone application on Tomcat, it must be reconfigured to fit your environment.

A WAR file is compressed using a compressed zip file. To access the configuration files, you must extract the zip file. Once the configuration files are updated, you re-zip and deploy ManagementConsole.war to your Tomcat server.

The table below contains a list of the configuration files relative to the root of the unzipped ManagementConsole.war.

Configuration Files

File	Configures	Notes
WEB-INF/Configuration.xml	Clustering, password encryption, REST-Plugin	If you copy the version of the file from 8.1, it will automatically be upgraded once you start the Management Console
WEB-INF/login.xml	Administrators and users, this is where you integrate with LDAP	
WEB-INF/classes/ log4j.properties	application logging	

Spring Configuration Files

Configuration.xml and login.xml are all Spring configuration files (www.springsource.org) and share the same general syntax which we will outline here.

Spring is configured through a series of beans, and each bean has properties that configure a piece of code inside the application. The general syntax:

```
<bean id="id" class="SomeClass">
   <property name="myName" value="myValue"/>
</bean>
```

File	Configures
`id="id"`	The id of the bean is an internal handle, that the application use to refer to the bean. It is also referred to as the bean's name.
`class="SomeClass"`	The class identifies the code component which the bean configures.
`<property name="myName" value="myValue"/>`	Defines a property with the name myName and the value myValue. This configures a property on the code component defined by the class attribute.

In Kapow versions prior to 9.3, user access credentials were defined manually in the login.xml file. Starting from version 9.3, user management is performed using the Users & Groups tab under the Admin tab in the Management Console. In the enterprise version of Kapow, user management is enabled by default. See the Users Tab topic for more details. User population from previous installations can be performed using the Kapow backup functionality. For LDAP integration you still need to edit the login.xml file.

Troubleshooting

If you have any problems during the installation, you should check the Tomcat log in the /logs folder in your Tomcat installation. During the configuration process it is often easier to run Tomcat from the command line, as it will then print error messages directly in the command line window.

Creating a New Database

We strongly recommend that you create a new database for the tables used by the Management Console. There are two requirements to the database.

- Unicode support
- Case-sensitive comparison

Unicode support is needed because non-ASCII characters, like Danish Æ, German ß or Cyrillic Ë may be given as input to robots. This input is stored in the database, and without Unicode support these characters may be stored incorrectly.

Case-sensitive comparison is needed because it is possible to upload a robot named a.robot and another named A.robot. Without case-sensitive comparison, uploading the latter would override the first.

Database servers handles Unicode and case-sensitive comparison very differently. The following list contains recommendations for the supported database systems.

Recommendations for Unicode Support and Case-Sensitive Comparison

Database	Recommendations
IBM DB2	Create the database using CODESET UTF-8
MySQL 5.x	Create the database with utf8_bin collation: `CREATE DATABASE KAPOW_MC COLLATE utf8_bin`
Oracle	We use NVARCHAR2, NCLOB types for Unicode. For case-sensitive comparison ensure that NLS_COMP is set to BINARY
Microsoft SQL Server 2005/2008	We use NVARCAR, NTEXT types for Unicode. For case-sensitive comparison create the database with a Case-Sensitive collation such as SQL_Latin1_General_CP1_CS_AS: `CREATE DATABASE KAPOW_MC COLLATE SQL_Latin1_General_CP1_CS_AS`

Database	Recommendations
Sybase Adaptive Server Enterprise	We use NVARCAR, NTEXT types for Unicode. For case-sensitive comparison ensure that the sort order is binary (see sp_helpsort). The scripts below use NVARCHAR(255) NOT NULL UNIQUE, due to index limitations this will not work on an ASE with 2K page size, unless you modify the scripts to use NVARCHAR(200) NOT NULL UNIQUE (or install on a 4/8/16K page server)

The tables used by the Management Console can be grouped into 3 categories: Platform tables, logging tables, and data view tables. The platform tables hold information exclusive to the Management Console such as the uploaded robots and their scheduling information, while the logging and data view tables are shared with RoboServer.

User Privileges

When the Management Console starts, it will automatically try to create the required platform tables and logging tables (if not already created by RoboServer). This means that the user account used to access the database must have the CREATE TABLE and ALTER TABLE privileges as well as the CREATE TEMPORARY TABLES privilege to restore a backup. Oracle users also need the CREATE SEQUENCE privilege. If this is not possible you can ask your Database administrator to create the tables using the scripts below.

Additionally the user must be allowed to SELECT, INSERT, UPDATE, DELETE for the system to work properly.

SQL Scripts for Management Console Tables

See Appendix for SQL scripts.

The Management Console uses a 3rd party scheduling component called Quartz. Quartz also requires a number of tables which must reside among the other platform tables. These tables are also created automatically when the Management Console starts, or may be created manually using the scripts in the Appendix.

Creating a Tomcat Context File

In an enterprise environment, databases are often accessed through a data source. This guide will show you how to configure your Tomcat with a data source that connects to a local MySQL database server.

In Tomcat, data sources are defined within the applications context. The context may be declared either embedded or external to the application. When the context is embedded, it is defined in the file context.xml, which must be located inside the WAR file inside the META-INF folder. When declared externally the file must be located in Tomcat's `/conf/Catalina/localhost` folder and the name of the file must be ManagementConsole.xml (same name as the deployed WAR file). Although Tomcat recommends deploying with an embedded context, as it provides a single deployment unit, we will use an external context definition in this guide, as it makes modifying the file easier. Once you have refined your configuration, you can embed the context file and deploy the War file to your production environment.

Adding Platform Data Source

Create the file ManagementConsole.xml inside Tomcat's/conf/Catalina/localhost folder and add the following content:

```
<Context useHttpOnly="true">
        <!-- Default set of monitored resources -->
        <WatchedResource>WEB-INF/web.xml</WatchedResource>

        <Resource name="jdbc/kapow/platform" auth="Container"
 type="javax.sql.DataSource"
               maxActive="100" maxIdle="30" maxWait="-1"
               validationQuery="/* ping */" testOnBorrow="true"
               username="MyUser" password="MyPassword"
 driverClassName="com.mysql.jdbc.Driver"
               url="jdbc:mysql://localhost:3306/KAPOW_MC?
useUnicode=yes&amp;characterEncoding=UTF-8&amp;rewriteBatchedStatements=true"/>

     </Context>
```

The url parameter above is a JDBC URL. The username and password attributes are used by Tomcat to create a connection pool used when connecting to the database.

The data sources are defined differently for other databases. For instance, if you are using Microsoft SQL Server 2005/2008, the relevant three lines above should instead be:

```
username="MyUser" password="MyPassword"
           driverClassName="com.microsoft.sqlserver.jdbc.SQLServerDriver"
           validationQuery="SELECT 1" testOnBorrow="true"
           url="jdbc:sqlserver://localhost:1433;DatabaseName=MyDbName"/>
```

Note that if you are using Microsoft SQL Server, you need to configure it to use mixed mode authentication. Generally, you should consult the JDBC documentation to identify which values to use in the data sources.

The URL `jdbc:mysql://localhost:3306/KAPOW_MC?useUnicode=yes&characterEncoding=UTF-8` refers to a database named KAPOW_MC in our local MySQL. For MySQL it is recommended that you add ?useUnicode=yes&characterEncoding=UTF-8 to all connection strings, otherwise the JDBC driver will not handle Chinese, Japanese or other 3-byte utf-8 characters correctly, since we can't have & directly inside the context xml file, we must encode it as &

rewriteBatchedStatements=true instructs the MySQL JDBC driver batch inserts/updates and should give improved insert performance for kapplet robots.

The driverClassName parameter controls which JDBC driver is used; each database vendor provides a JDBC driver for their database, which you will have to download. The JDBC driver, typically a single .jar file, must be copied into the /lib folder on Tomcat 6/7, or commons/lib on Tomcat 5.5.

The validationQuery is used by Tomcat to verify that the connection obtained from the connection pool is still valid (as the database server may have closed the connection). The validation query is lightweight and uses very few resources on the database server, this list contains validation queries for the supported databases.

Validation Queries

Database	Query
MySQL	/* ping */

Database	Query
Microsoft SQL Server 2005/2008	SELECT 1
Sybase Adaptive Server Enterprise	SELECT 1
IBM DB2	VALUES(1)
Oracle	SELECT 1 FROM DUAL

Note that the MySQL JDBC driver supports a special lightweight /* ping */ 'request', check JConnector manual section 6.1 for details

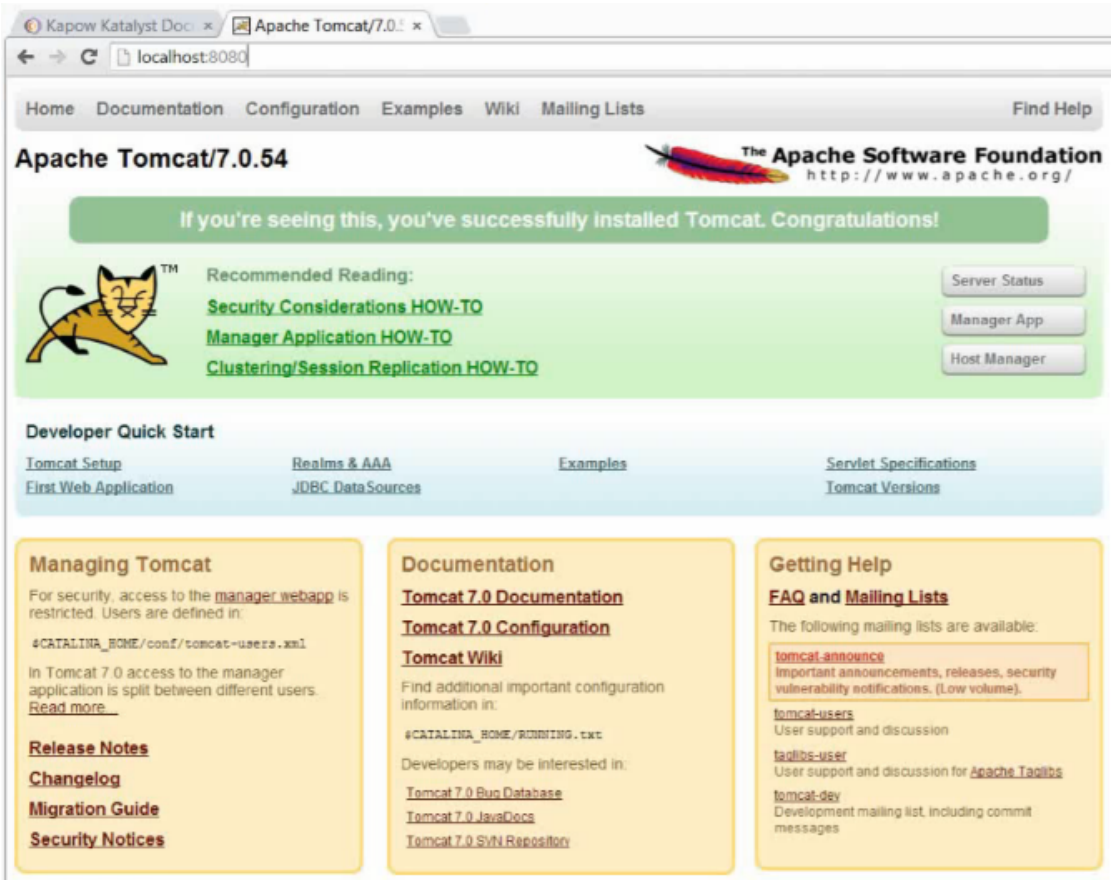
For more information on context configuration and data sources, see JNDI Resources HOW-TO and JNDI data source HOW-TO.

We are now ready to start the Tomcat server.

Starting Tomcat

Start your Tomcat server, wait a couple of seconds for the application to be deployed, then navigate to http://localhost:8080/. You should see the following page.

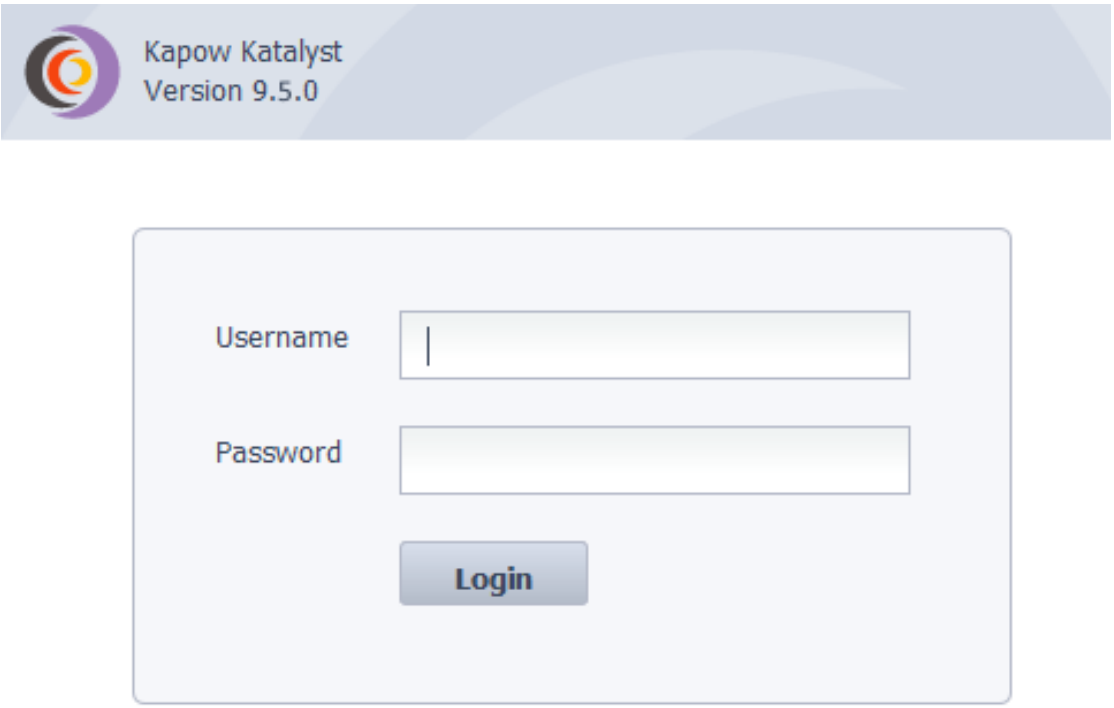
No Providers Apache Start Page



By default there is only one set of credentials (user name - `admin`, password - `admin`) with administrator privileges. Later you can change password and create other users and groups on the Users Tab under the Admin tab in the Management Console.

Now open the http://localhost:8080/ManagementConsole you should see the following login screen.

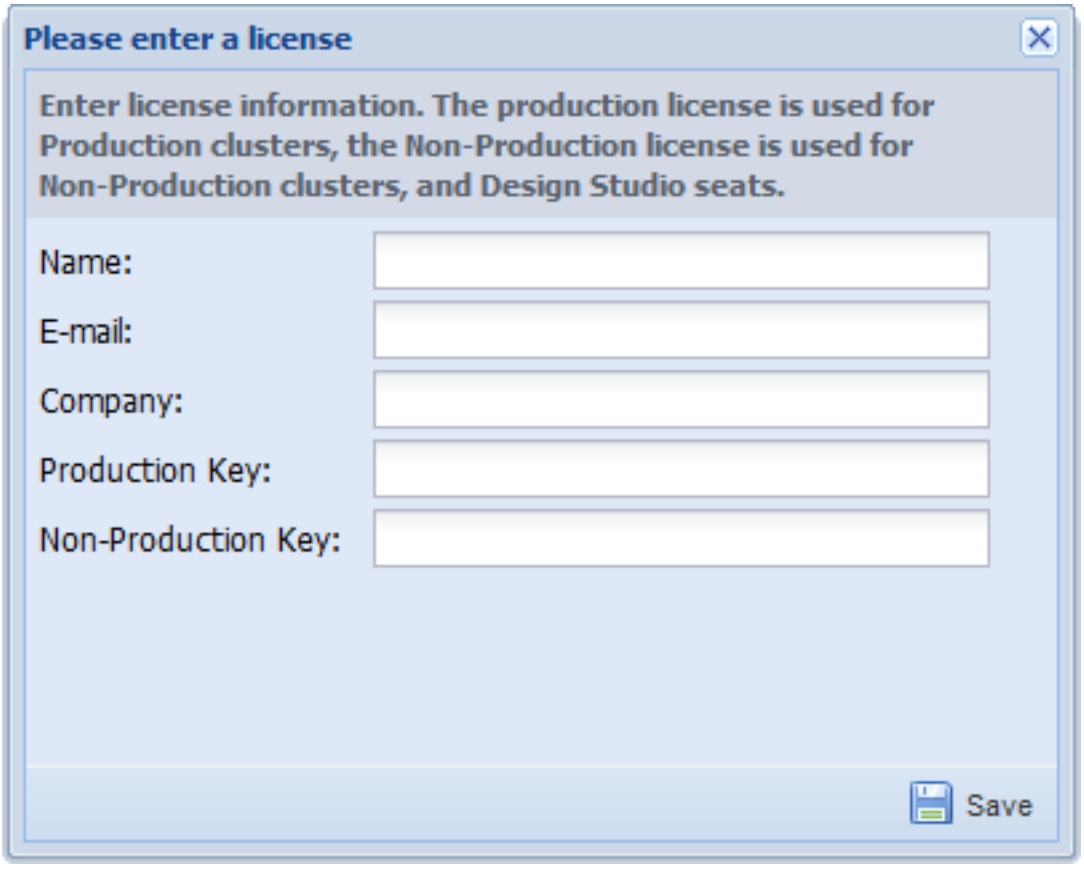
Login Page



Enter `admin` as the username and `admin` as the password and click **Login**.

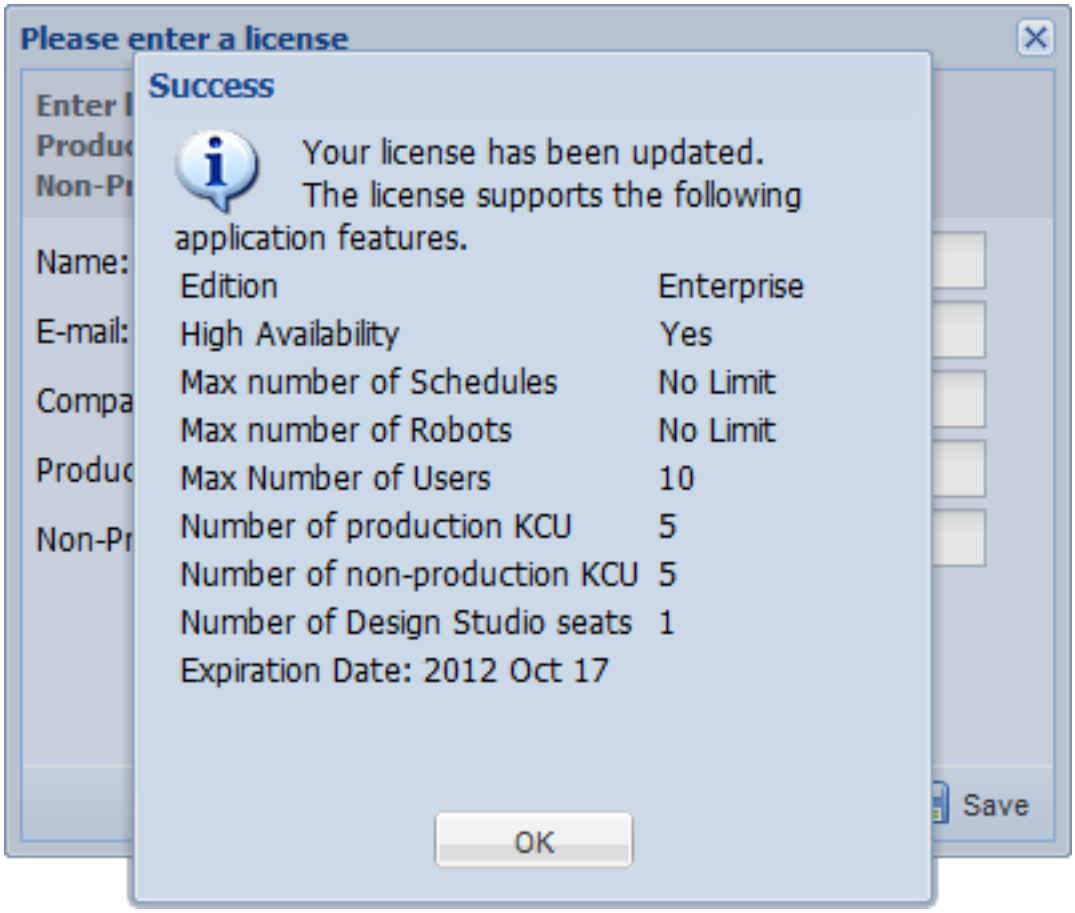
Entering License Information

After logging in, the license window is displayed.



Now enter the Kapow Katalyst license information and click **Save**. You should see the following dialog box displaying which features enabled by your license key.

License Updated



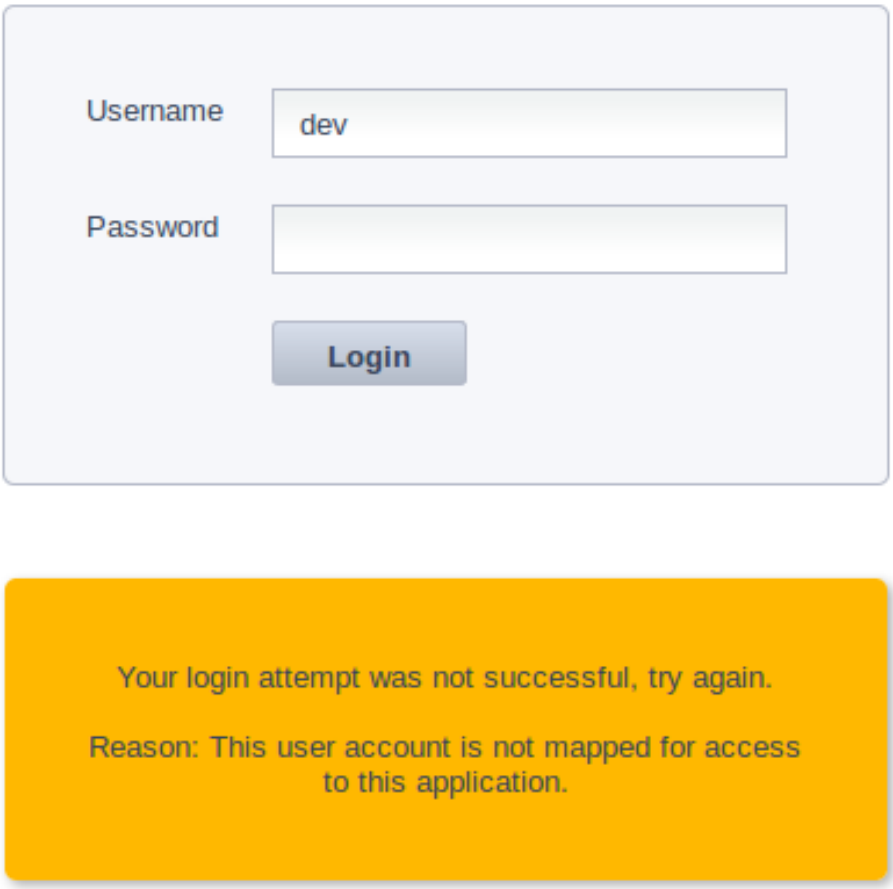
Project Permissions

The admin user account used to log in bypasses the normal project permissions applied to regular users, because admin is the superuser. The superuser can not be a member of any group, and has an unrestricted access to all projects.

To change the admin password and create new users and groups, go to the Admin tab and then click the **Users and Groups** tab. The security model is role-based; after you create a user, you must add this user to one or more groups associated with specific roles in one or more projects.

Create DEVELOPER group, then create a user dev and add this user to the DEVELOPER group. Try to log in with this user credentials. You should see the following screen:

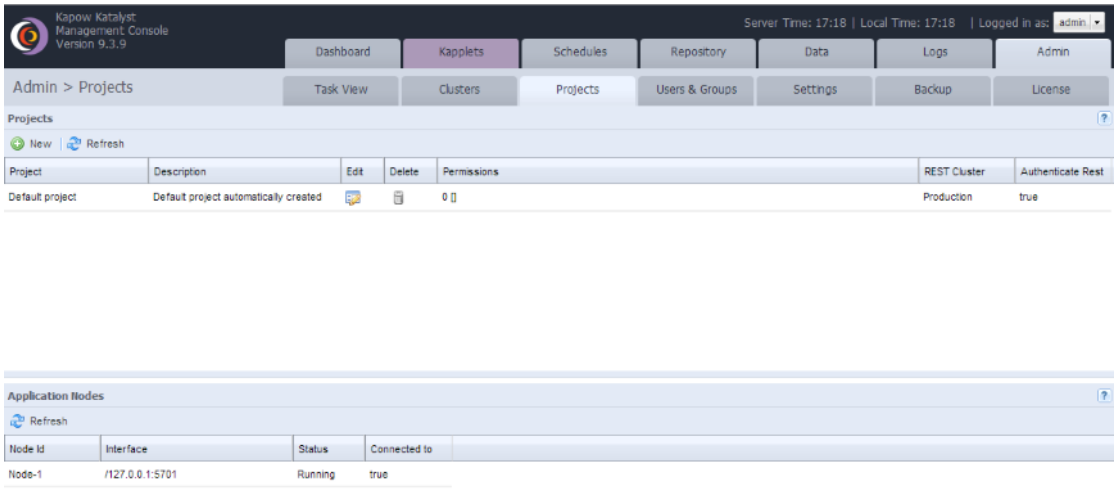
User Not Mapped



For non-administrator users, permissions are based on the user's group membership. The dev user is a member of the group DEVELOPER, and since we have not given users in the DEVELOPER group access to any projects, they are not able to log in.

To allow the dev user to log in, we must log in as administrator and go to the Projects tab which is a sub tab of the Admin tab. It looks like this:

Initial Project Permissions

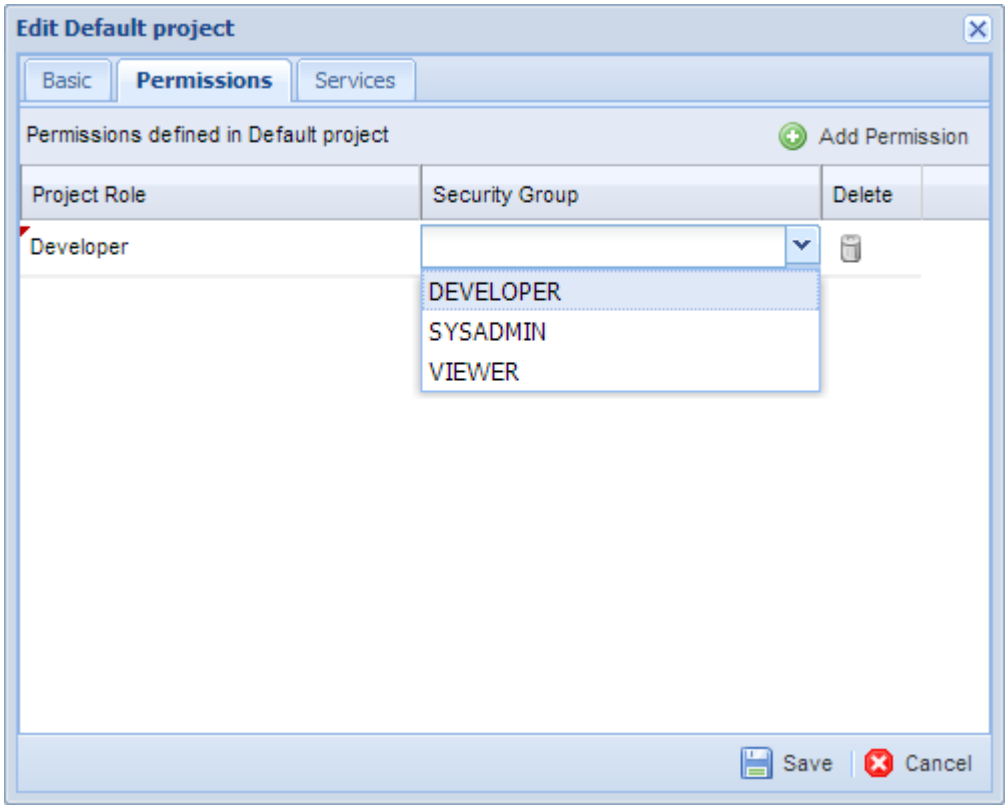


In the Permission column, there are 0 permission mappings.

Click Edit to open the Edit Project dialog box and go to the Permissions tab. There is a grid with columns Project Role, and Security Group. A project role determines a set of actions that may be performed inside the Management Console, such as uploading robots, creating schedules, viewing logs etc. Within a project you assign a project role to a security group. That way, all users of the selected security group will be able to perform the actions allowed by the assigned project role.

Click Add Permission to add permissions in this project. This adds a new line to the grid, and inserts a dropdown box allowing us to select a project role. Select the project role developer. Now double click in the Security Group column and select the DEVELOPER security group (of which our dev user is a member). It should look like this:

Adding Project Permissions

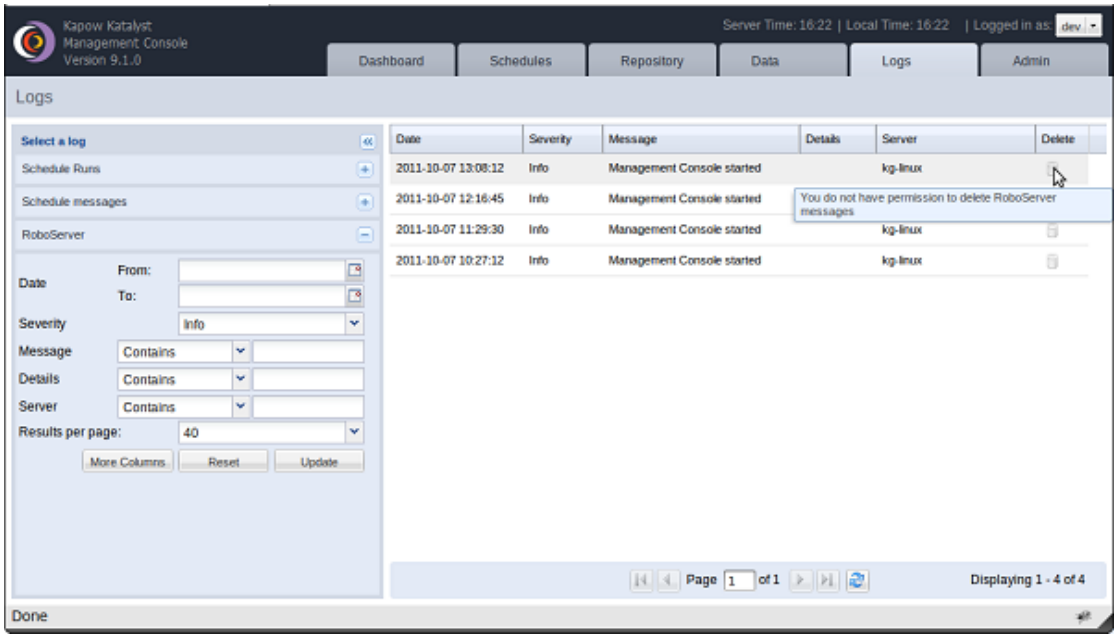


Now click Save.

All members of the DEVELOPER group can now perform the actions allowed by the role developer.

Let's log in as the dev user and see how the permissions are reflected in the Management Console. You log out by clicking the menu button in the upper right corner, then log in as the dev user. Now go to the Logs tab, select the RoboServer log in the left pane. Notice how the delete button is disabled, and hovering gives a tooltip message that you do not have permissions to delete RoboServer messages.

View Logs But Not Delete



You can assign multiple roles to the same security group, and you can assign the same role to multiple security groups. If a user holds multiple roles, he can do anything that at least one of the roles allow. With multiple projects in Management Console, users of different projects can be completely separated by assigning their groups to project-specific roles.

The predefined roles are suggestions, but you can add any number of additional roles, or change the existing roles to fit your needs.

Actions that can be performed on the Settings, Backup and License tabs (sub tabs to Admin) are only available to users that are members of the Administrator group.

For using your LDAP user accounts, see the Advanced Configuration>LDAP Integration topic.

Security

In the WEB-INF/Configuration.xml file, it is possible to configure some additional security settings for the Management Console.

The security configuration section of the file looks like this:

```
<bean id="securityConfiguration"
 class="com.kapowtech.mc.config.SecurityConfiguration">
   <property name="allowScriptExecution" value="false"/>
   <property name="jdbcDriverUpload" value="LOCALHOST"/>
</bean>
```

It contains two security settings which are described below.

As part of a schedule, a user may configure a script that is used for Pre/Post processing, e.g. before/after any robots in the schedule are executed. These scripts must be located on the Management Console computer's file system. For security reasons, the running of external

scripts is disabled by default. To allow external scripts to execute, you should change the value of the allowScriptExecution property to true.

If you use a pre/post-processing script in a schedule, and scripts are disabled by the administrator, the schedule will fail immediately with the following message Scripts have been disabled by the administrator. without executing any robots. Users can save schedules with pre/post-processing scripts even if the feature is disabled. NB: Even if external scripts are disabled, you can still use the pseudo script command “runrobot: test.robot” as part of pre- or post-processing.

By default, only the admin user when accessing the Management Console from the localhost is allowed to upload JDBC drivers. To change this behavior, modify the jdbcDriverUpload property. The following are the possible values, the property can have:

- NONE: upload of JDBC drivers is not allowed for any users
- LOCALHOST: the default value, the admin user is allowed to upload drivers if accessing the Management Console from the localhost
- ANY_HOST: the admin user is allowed to upload drivers from any host

Deployment Checklist

Use the following checklist to ensure that all deployment tasks are completed in the proper order.

Deployment Checklist

Item	Description
Download and install a supported version of Apache Tomcat Server	If your setup requires access to the Management Console outside of your corporate intranet, make sure SSL is set up to work with your Tomcat server.
Download and install Java 8 (JDK 1.8.0_40 or later)	Management Console will not initialize correctly if Apache Tomcat is started using any version of Java other than Java 8.
Ensure that the JAVA_HOME variable is pointed to the Java 8 installation.	
Start up the Apache Tomcat and confirm that the Apache Tomcat server will come online before attempting to deploy the Management Console application.	▪ You can use the catalina run command from the \bin directory to start the server. ▪ Going to http://localhost:8080 in the browser should display the default Apache Tomcat start up page.
From a Kapow 9.3.x installation, locate the ManagementConsole.war file that will be deployed under the Apache Tomcat \webapps directory	Expect errors on the initial start up of the application at this point, because the server has not been configured yet. We are simply unpacking the .war file so that we can easily gain access to the files we will need to edit to complete the installation and configuration process.

Item	Description
Turn the Apache Tomcat server off.	
Create a new database to be used by the Enterprise Management Console to hold its configuration.	▪ Select one of the support platforms. ▪ The online help contains the CREATE and DROP SQL Scripts needed to create all of the required DB tables. Note that these scripts only need to be used if the user account that the application will use to connect to the DB does not have CREATE privileges on the schema. ▪ Select an appropriate collation as suggested in the on-line help.
Create the DB user account to be used by the application to connect to the database via JDBC.	
Create the Tomcat Context file: ManagementConsole.xml	▪ Validation query to be specified is DB platform specific. The on-line help has all accepted values for all supported databases. ▪ Do not forget to update the Username, Password, and DatabaseName parameters in the JDBC URL string with the correct values for your database.
Prior to starting up the application, notice that the Management Console database (as specified in the ManagementConsole.xml) does not have any tables related to the Management Console process at this time.	▪ When the application is started the needed DB tables will be automatically created if they are not present assuming that the DBUser account has CREATE privileges to the schema. ▪ If the user does NOT have CREATE privileges, the CREATE SQL scripts can be downloaded from the help.kapowsoftware.com website.
Do not forget to deploy the necessary JDBC .jar file under the Apache Tomcat installation directory.	▪ Deploying to <APACHE_TOMCAT_INSTALL_DIR>\lib makes it available to ALL apps running under Tomcat. ▪ Deploying to <APACHE_TOMCAT_INSTALL_DIR>\webapps \ManagementConsole\WEB-INF\lib makes the JDBC .jar file only accessible to the Management Console application itself.
Restart the Apache Tomcat Server	Confirm the Tomcat main home page loads: http://localhost:8080
Try to login to the Enterprise Management Console as user - admin, password - admin	http://localhost:8080/ManagementConsole
Enter the Kapow Software License Keys	

Advanced Configuration

LDAP Integration

The installation guide shows you how to authenticate users against a list of hard-coded credentials, and this section describes how to authenticate users against LDAP.

In login.xml, you will find the following definition:

```
<bean id="ldap" class="com.kapowtech.mc.config.LdapLogin" lazy-init="true">
        <property name="ldapServerURL" value="ldap://change-to-ldapHost:389"/>
```

```
        <property name="userDn" value="CN=LDAP test,CN=Users,DC=kapowdemo,DC=local"/>
        <property name="password" value="change-to-passowrd"/>
        <property name="userSearchBase"
value="OU=Users,OU=TheEnterprise,DC=kapowdemo,DC=local"/>
        <property name="userSearchFilter"
value="(userPrincipalName={0}@kapowdemo.local)"/>
        <property name="userSearchSubtree" value="true"/>
        <property name="groupSearchBase" value="OU=Security
Groups,OU=TheEnterprise,DC=kapowdemo,DC=local"/>
        <property name="groupSearchFilter" value="(member={0})"/>
        <property name="groupRoleAttribute" value="cn"/>
        <property name="groupSearchSubtree" value="true"/>
        <property name="convertToUpperCase" value="true"/>
        <property name="allGroupsFilter" value="cn=E*"/>
        <property name="fullNameAttribute" value="displayName"/>
        <property name="emailAttribute" value="userPrincipalName"/>
</bean
```

This defines an LdapLogin bean named ldap. The bean defines a number of properties that controls the LDAP integration. If you are familiar with the way Tomcat integrates to LDAP this should be quite familiar.

Deployment Checklist

Property	Description
ldapServerURL	The URL to the LDAP server. This uses the ldap:// protocol.
userDn	The DN (distinguished name) used to log in to LDAP to authenticate other users.
password	The password for the userDN account. As the password will be stored in clear text in this file you should use an account that only has 'read' access.
userSearchBase	Subdirectory in the LDAP tree where users can be found.
userSearchFilter	Filter that is applied to find the username.
userSearchSubtree	Set to true if users may be located in the subdirectory of the userSearchBase.
groupSearchBase	Subdirectory in the LDAP tree where groups can be found.
groupSearchFilter	Filter that is applied to identify the users in this group.
groupRoleAttribute	Attribute that holds the group name.
groupSearchSubtree	Set to true if groups may be located in the subdirectory of the groupSearchBase
convertToUpperCase	Should the group names be converted to upper case, true by default.
allGroupsFilter	Optional. Controls which groups are displayed when creating project permissions, see below.
fullNameAttribute	Attribute to fetch the full name of the user.
emailAttribute	Attribute to fetch the email of the user.

To use an LDAP account to administer the Management Console, you must add one of the groups that you are are a member of to the adminGroups bean in login.xml, as described in Project Permissions. Be advised that anyone that is a member of a group listed in adminGroups will be a Management Console administrator, so you may want to create a new LDAP group for this purpose. Take care to use the upper case group name if convertToUpperCase is true.

When you select a project permission, you will see that all the group names have been pulled from LDAP to populate the drop down. The groups are located by using the

groupRoleAttribute to construct a filter to fetch all groups. Sometimes you don't want all LDAP groups displayed here, in which case override this behavior by providing your own filter, this is done by adding an additional property to the LdapLogin.

```
<property name="allGroupsFilter" value="(cn=*)"/>
```

will find all group names, if the group name is in the cn attribute (this is the default). If you only want groups starting with the letter 'e' you could use

```
<property name="allGroupsFilter" value="(cn=E*)"/>
```

The filter uses basic LDAP queries, so you can find documentation elsewhere for more complex queries.

High Availability

If high availability (failover) is required you can configure multiple Management Console instances to work together as a cluster. Four components must be clustered to achieve full failover.

Cluster Components

Component	Description
Load balancer	An HTTP load balancer is required to distribute requests between multiple Tomcat servers.
Clustered platform database	The Management Console stores schedules, robots etc. in the platform database. In a failover scenario the platform database should run on a clustered DBMS to avoid a single point of failure.
Tomcat session replication	Although the Management Console doesn't store any data directly in the user's session (except during Import/Export), the session holds the user's authentication information. If session replication is not enabled, the user will have to login again if the Tomcat he is currently connected to crashes.
Hazelcast	Hazelcast (www.hazelcast.com) is used to cluster data structures over multiple JVMs. Inside the Management Console this is used to provide clustering of vital data structures, and to provide intercommunication between application instances. Here is a example: When you run a robot on RoboServer, a thread is required to process the status messages returned by RoboServer. This thread will be running inside a concrete Tomcat instance. In a clustered environment, a user trying to stop the robot may in fact be generating the stop request on another Tomcat instance than the instance running the robot. In that case the stop request is broadcast through Hazelcast to all instances and the instance running the robot will receive it and act to stop the robot.

Multiple Management Console Instances

You should have two or more identical Tomcat installations, and deploy the same version of ManagementConsole.war on them all. Make sure the web.xmlConfiguration.xml, login.xml and roles.xml files are the same across all the instances.

High Availability

If high availability (failover) is required you can configure multiple Management Console instances to work together as a cluster. Four components must be clustered to achieve full failover.

Cluster Components

Component	Description
Load balancer	An HTTP load balancer is required to distribute requests between multiple Tomcat servers.
Clustered platform database	The Management Console stores schedules, robots etc. in the platform database. In a failover scenario the platform database should run on a clustered DBMS to avoid a single point of failure.
Tomcat session replication	Although the Management Console doesn't store any data directly in the user's session (except during Import/Export), the session holds the user's authentication information. If session replication is not enabled, the user will have to login again if the Tomcat he is currently connected to crashes.
Hazelcast	Hazelcast (www.hazelcast.com) is used to cluster data structures over multiple JVMs. Inside the Management Console this is used to provide clustering of vital data structures, and to provide intercommunication between application instances. Here is a example: When you run a robot on RoboServer, a thread is required to process the status messages returned by RoboServer. This thread will be running inside a concrete Tomcat instance. In a clustered environment, a user trying to stop the robot may in fact be generating the stop request on another Tomcat instance than the instance running the robot. In that case the stop request is broadcast through Hazelcast to all instances and the instance running the robot will receive it and act to stop the robot.

Multiple Management Console Instances

You should have two or more identical Tomcat installations, and deploy the same version of ManagementConsole.war on them all. Make sure the web.xmlConfiguration.xml, login.xml and roles.xml files are the same across all the instances.

Tomcat Session Replication

Session replication is configured in /conf/server.xml. Here is an example that uses multicast for instance discovery (Tomcat 5.5).

```
<Cluster className="org.apache.catalina.cluster.tcp.SimpleTcpCluster"
         managerClassName="org.apache.catalina.cluster.session.DeltaManager"
         expireSessionsOnShutdown="false"
         useDirtyFlag="true"
         notifyListenersOnReplication="true"
         printToScreen="true">

        <Membership
            className="org.apache.catalina.cluster.mcast.McastService"
            mcastAddr="228.0.0.4"
            mcastPort="45564"
            mcastFrequency="500"
            mcastDropTime="3000"/>
```

```
        <Receiver
            className="org.apache.catalina.cluster.tcp.ReplicationListener"
            tcpListenAddress="auto"
            tcpListenPort="4002"
            tcpSelectorTimeout="100"
            tcpThreadCount="6"/>

        <Sender
            className="org.apache.catalina.cluster.tcp.ReplicationTransmitter"
            replicationMode="pooled"
            ackTimeout="150000"
            waitForAck="true"/>

        <Valve className="org.apache.catalina.cluster.tcp.ReplicationValve"
               filter=".*\.gif;.*\.js;.*\.jpg;.*\.png;.*\.htm;.*\.html;.*\.css;.*
\.txt;"/>

        <Deployer className="org.apache.catalina.cluster.deploy.FarmWarDeployer"
                  tempDir="/tmp/war-temp/"
                  deployDir="/tmp/war-deploy/"
                  watchDir="/tmp/war-listen/"
                  watchEnabled="false"/>

        <ClusterListener
 className="org.apache.catalina.cluster.session.ClusterSessionListener"/>
    </Cluster>
```

You also have to set the jvmRoute attribute on the <Engine> element in server.xml:

```
<Engine jvmRoute="tomcat2" name="Catalina" defaultHost="MyHost">
```

Note If you are using mod_jk as a poor man's load balancer, the value of the jvmRoute has to match the name listed in the workers.properties file references by the mod_jk configuration.

See your Tomcat documentation for details.

Hazelcast Replication

The most basic Hazelcast settings can be edited in Configuration.xml, while more advanced settings such as SSL encryption must be configured in /WEB-INF/Hazelcast.xml

When Management Console starts, it creates a Hazelcast node on port 5701 (or the next available port that is available). By default this Hazelcast node will bind to IP address 127.0.0.1. You will have to change the bind address to a public IP/host name before it can participate in a cluster. This is done by modifying the interface property of the cluster bean in Configuration.xml. It might look like this:

```
    <bean id="cluster" class="com.kapowtech.mc.config.ClusterConfig" >
        <property name="port" value="26000"/>
        <property name="interface" value="10.0.0.*"/>
    .......
    </bean>
```

The * is used as a wildcard, in this case the application will try bind to the 'first' interface that has an IP address starting with 10.0.0. It is possible, but not recommended to use *.*.*.* as you may end up binding to 127.0.0.1, or another virtual interface.

When you start additional instances of Management Console, their Hazelcast instances will try to find any existing Hazelcast node and join the cluster. This discovery can be done through multicast or through TCP/IP.

To use multicast discovery you must modify the cluster bean in Configuration.xml. This is done my un-commenting the following line:

```
<property name="joinConfig" ref="multicastCluster"/>
```

multicastCluster is a reference to the multicastCluster bean, which defines the multicast group and port. You may change it to fit your network topology.

If your network doesn't allow multicast you will have to use the tcpCluster. That is done by un-commenting this line instead:

```
<property name="joinConfig" ref="tcpCluster"/>
```

The tcpCluster bean contains a list of TcpPeer, one for each other Hazelcast node. If you use the same TCP port for all Hazelcast nodes you don't need to specify a port number (each node will assume that its peers are running on the same port as itself). If you have two nodes configured in a TCP cluster it could look like this:

```
<bean id="tcpCluster" class="com.kapowtech.mc.config.TcpJoinConfig">
     <property name="peers">
         <list>
             <bean class="com.kapowtech.mc.config.TcpPeer">
                 <property name="host" value="10.0.0.25"/>
             </bean>
             <bean class="com.kapowtech.mc.config.TcpPeer">
                 <property name="host" value="10.0.0.26"/>
             </bean>
         </list>
     </property>
 </bean>
```

Notice that both nodes are in the list. This means that regardless which node starts first it will be able to find its peer. It also allows you to use identical Configuration.xml files in both applications. Also, TCP ports numbers are not defined, so each peer will try to connect to the other one on the same port as it is listening on itself.

Application Nodes

You can verify that the application is properly clustered by going to the Projects tab, and look at the Application Nodes section in the bottom of the page. Here you should see something like this:

INSERT SCREEN SHOT

This means that the cluster currently consists of two nodes. The interface column will list the IP/host and port that Hazelcast is using for inter-cluster communication. The Connected to column informs you which of the two nodes you are connected to at the moment. If you shut down the server you are currently connected to, you will automatically be re-routed to another live instance by the load balancer.

If you right click on an application node a context menu will appear, here you can request a thread dump from any node, this may be useful for debugging purposes.

URI Encoding

If you plan to upload robots with names that contain non-ASCII characters, like Danish ÆØÅ or German ß to the repository, you have to configure the URI Encoding on your web container to UTF-8.

On Tomcat this is done on the <connector> definition found in server.xml file inside the /conf folder. Here you add the attribute URIEncoding="UTF-8" like this:

```
<Connector port="8080" URIEncoding="UTF-8"...../>
```

Password Encryption

As of version 8.2 the Management Console uses certificate based (public-, private-key) encryption when storing passwords. When you import from a previous version password will automatically be re-encrypted using the new certificate based algorithm.

The certificate and the matching private key is stored in a Java keystore, Management Console ships with a keystore that contains a default certificate and private key. Since all customers get the same keystore we recommend that you create your own keystore, otherwise anyone will be able to load your exports and potentially get your passwords.

Create Your Own Keystore

If you have already started the Management Console you will need to upgrade the certificate. The keystore must be in pkcs12 format, and can be created using the keytool application that comes with the Java SDK (which can be downloaded from Oracle.com, currently available here). The following command creates a new pkcs12 keystore with a certificate that is valid for 365 days.

```
keytool -genkey -alias mc -keyalg RSA -validity 3650 -keystore mc.p12 -
storetype pkcs12
```

You will be prompted for password, and the information that will be stored in the X.509 private key. The command will create a file mc.p12 (the value from the -keystore argument) in the current directory. -validity 3650 means the certificate will be valid for 10 years.

Note We don't recommend that you use a certificate issued by a certificate authority (CA) since pkcs12 holds both the private key and the public certificate, and the password to the private key will be written in clear text as part of the application configuration.

To instruct Management Console to use the new certificate, change the Configuration.xml file. The file is located inside the ManagementConsole.war web archive, which must be unpacked, see Deploying into Tomcat for details. Inside Configuration.xml you will find the following entry:

```
<bean id="keyStore" class="com.kapowtech.mc.config.KeyStoreConfig" >
      <property name="location" value="/WEB-INF/mc.p12"/>
      <property name="password" value="changeit"/>
      <property name="alias" value="mc"/>
</bean>
```

Here you must specify the location, password and alias of the keystore. If you copy the keystore into ManagementConsole.war the location must be relative to the root of the application. If you want to refer to a keystore stored in the file system, the location must start with file://, and must be an absolute reference to the keystore location.

Upgrading the Keystore

The first time Management Console starts, it creates a checksum using the private key from the keystore, this allows it to detect when the keystore has been replaced, and verify that passwords can in fact be decrypted with the provided certificate. If you have already started Management Console before installing your own keystore, you have will have to configure Management Console to perform a password conversion.

First copy the current keystore file into a new location, like your users home folder, then modify Configuration.xml to create a password converter with reference to the old keystore:

```
      <bean id="oldKeyStore" class="com.kapowtech.mc.config.KeyStoreConfig" >
         <property name="location" value="file:///home/roboserver/mc.p12"/>
         <property name="password" value="changeit"/>
         <property name="alias" value="mc"/>
      </bean>

      <bean id="passwordConverter"
 class="com.kapowtech.scheduler.server.service.PasswordConverter">
         <constructor-arg ref="oldKeyStore"/>
      </bean>
```

This configures a password converter to use the previous certificate to decrypt any existing passwords and checksum (you will have to provide correct location, alias and password for the old keystore), and use the new private key (as configured above) to re-encrypt passwords and create a new checksum. The conversion will occur the next time the Management Console is started, the conversion occurs while the application is starting and may take some time if there are many schedules. You don't have to remove the oldKeyStore and passwordConverter beans from Configuration.xml, as the password conversion is only triggered when the checksum and keystore is out-of-sync, and after the conversion the checksum will match the new keystore).

SSL Endpont Verification

When you create a new Cluster you can select that you want the communication with the RoboServers to be SSL encrypted, this prevents anyone from “listening” to the network and extracting critical information exchanged between the two parties.

In addition to encryption, SSL also offer endpoint validation. This is to ensure that you don't exchange critical information with a third party, either due to misconfiguration, or because you DNS has been hacked. For this to work you need to configure RoboServer to trust your Management Console and configure Management Console to trust your RoboServers.

This requires you to edit files inside ManagementConsole.war, so make sure you Tomcat server is not running when you perform this modification.

Certificates

You will need to create two certificates, one for Management Console and one for RoboServer, each certificate contains a private and a public key. Creating a certificate and exporting the public key is described here, in general it is a good idea to read the entire section of the help the discusses certificates, especially the section on API Client/Server certificates.

Endpoint verification can be separated into two parts, making RoboServer trust Management Console and making Management Console trust RoboServer, each of these are configured individually, and you don't have to configure both.

Making RoboServer Trust Management Console

You now have to tell Management Console to use the private key when creating the SSL connection to RoboServer. This is done by modifying /WEB-INF/certs.xml found inside the WAR file. Here you will need to provide the location, and the password for the certificate, which could look like this:

```
<bean id="sslCertificationConfiguration"
 class="com.kapowtech.mc.config.SSLVerificationConfiguration">
                  ...
     <property name="privateCertificateLocation" value="file:///home/roboserver/
client.p12"/>
```

```
    <property name="privateCertPassword" value="changeit"/>
</bean>
```

Management Console is now using it private key when establishing SSL connections, once Management Console's public key is deployed in RoboServer's /TrustedClients folder it will allow the RoboServer to verify that it is connected to the right Management Console. You can read about the /TrustedClients here Remember to enable Verify API Client Certificates in RoboServer Settings, and deploy the public key on all RoboServers in the cluster.

Making Management Console Trust RoboServer

RoboServer already comes with a API certificate installed, so you have to create a new certificate and replace the pre-configured. First create the certificate as described above, then start RoboServer Settings and go to the Certificates tab, and click the change button, select the certificate, and enter the password when prompted. RoboServer will now use the new certificate when creating SSL connection with Management Console (and other API clients).

Now you need to configure Management Console to only trust SSL connections from RoboServers with the correct certificate. Like Management Console's client certificate this is (partly) configured in /WEB-INF/certs.xml, using the following two options:

```
<bean id="sslCertificationConfiguration"
class="com.kapowtech.mc.config.SSLVerificationConfiguration">
    <property name="verifyRoboServerCert" value="true"/>
    <property name="checkHostName" value="true"/>
    ...
</bean>
```

The option for verifying RoboServer certificates is a simple boolean flag (true/false), this is because you have to import the RoboServers public key into the JRE's default keystore. The JRE's default keystore is a file named cacerts located at /jre/lib/security/.

To import RoboServer's public key into `cacerts`, you use use the following command:

```
keytool -import -alias RoboServer -keystore cacerts -trustcacerts -file
server.pub.cer
```

You will be prompted for a password, which is changeit unless you have changed it. The alias will have to be unique, so if you created a separate certificate for each RoboServer, you will have to add a suffix. Also note that the references to cacerts and server.pub.cer are relative in this example.

The checkHostName option ensures that Management Console will only talk to the RoboServer if it presents the correct certificate AND is contacted using the hostname written inside the RoboServer's certificate. Note that localhost and 127.0.0.1 is not considered the same host when it comes to hostname checks.

Troubleshooting

Troubleshooting can be quite hard as there is virtually no information available if SSL connections can't be established, but it is important to know that:

- Management Console will not start if it can't find the certificate, or if the password is wrong.
- When you change RoboServer's certificate in RoboServer settings, it checks that the password is correct before storing the certificate.

If Management Console can't connect to a RoboServer the following may help you troubleshoot:

- Is RoboServer running. Try to telnet to the socket to be sure.
- Is the RoboServer host name correct (if checkHostName is enabled).
- Is the v public key imported into cacerts. Use keytool -list -v -keystore cacerts -alias RoboServer if you give -alias it lists all certificates.
- Has Management Console s public certificate been copied to RoboServers /TrustedClients folder.
- Check expiration date. The public key contains the expiration data of the private key, and can be opened/view in both Windows and Linux.

SQL Scripts for Creating Tables

SQL Scripts for Management Console Tables

Database	Create Tables	Drop Tables
IBM DB2 **Note** The database must have a "table space" with a page size of at least 8192 KB to create all tables.	CREATE TABLE MC_SCHEDULE (ID BIGINT NOT NULL, BAD_INPUT SMALLINT DEFAULT 0, ACTIVE SMALLINT DEFAULT 0, CREATEDBY VARCHAR(255), DIRTY SMALLINT DEFAULT 0, EMAILS VARCHAR(255), MAXOBJECTSEXTRACTED INTEGER, MAXRUNTIME INTEGER, MODIFIEDBY VARCHAR(255), NAME VARCHAR(255) NOT NULL, NEXTRUN TIMESTAMP, PREVIOUSRUN TIMESTAMP, TOTALRUNS INTEGER, USEEMAILNOTIFICATION SMALLINT DEFAULT 0, CLUSTER_ID BIGINT, P_ID BIGINT NOT NULL, BLOCKJOB_ID BIGINT, PRIMARY KEY (ID)); CREATE TABLE MC_BLOCK (ID BIGINT NOT NULL, DTYPE VARCHAR(31), NAME VARCHAR(255) NOT NULL, NAMEDBLOCK SMALLINT DEFAULT 0, P_ID BIGINT NOT NULL, PRIMARY KEY (ID)); CREATE TABLE MC_BLOCK_REFERENCE (ID BIGINT NOT NULL, ENABLED SMALLINT DEFAULT 0 NOT NULL, ORDER_VAL INTEGER NOT NULL, NAME VARCHAR(255) NOT NULL, CHILD BIGINT, PARENT BIGINT NOT NULL, PRIMARY KEY (ID)); CREATE TABLE MC_BLOCK_JOB (ID BIGINT NOT NULL, ACTIVE SMALLINT DEFAULT 0 NOT NULL, BLOCKINPUT CLOB(268435456), DISPLAY_NAMES CLOB(268435456) NOT NULL, BLOCK_ID BIGINT NOT NULL, PRIMARY KEY (ID)); CREATE TABLE MC_ROBOT_BLOCK (ID BIGINT NOT NULL, ROBOT_ID BIGINT, PRIMARY KEY (ID)); CREATE TABLE MC_MULTIPLE_ROBOT_BLOCK (ID BIGINT NOT NULL, DISPLAY_NAME VARCHAR(255), STRATEGY VARCHAR(255), VALUE VARCHAR(255), PRIMARY KEY (ID)); CREATE TABLE MC_CONTAINER_BLOCK (ID BIGINT NOT NULL, MAPPING BLOB(268435456), PRIMARY KEY (ID)); CREATE TABLE MC_SEQUENTIAL_BLOCK (ID BIGINT NOT NULL, PRIMARY KEY (ID)); CREATE TABLE MC_CONCURRENT_BLOCK (ID BIGINT NOT NULL, PRIMARY KEY (ID)); CREATE TABLE MC_SCRIPT_BLOCK (ID BIGINT NOT NULL, SCRIPT VARCHAR(255), PRIMARY KEY (ID)); CREATE TABLE MC_FOLDER_ITEM (ID BIGINT NOT NULL, DTYPE VARCHAR(31), FOLDER BIGINT NOT NULL, PRIMARY KEY (ID)); CREATE TABLE MC_ROBOT (ID BIGINT NOT NULL, BYTESIZE INTEGER NOT NULL, CONTAINSRETURNOBJECT SMALLINT DEFAULT 0, CREATEDBY VARCHAR(255), LASTMODIFIED TIMESTAMP NOT NULL, MODIFIEDBY	ALTER TABLE MC_SCHEDULE DROP CONSTRAINT MCSCHDULEBLCKJOBID; ALTER TABLE MC_SCHEDULE DROP CONSTRAINT MC_SCHEDULE_P_ID; ALTER TABLE MC_SCHEDULE DROP CONSTRAINT MCSCHEDULECLSTERID; ALTER TABLE MC_SCHEDULE DROP CONSTRAINT UNQ_MC_SCHEDULE_0; ALTER TABLE MC_BLOCK DROP CONSTRAINT FK_MC_BLOCK_P_ID; ALTER TABLE MC_BLOCK DROP CONSTRAINT UNQ_MC_BLOCK_0; ALTER TABLE MC_BLOCK_REFERENCE DROP CONSTRAINT MCBLCKRFERENCECHLD; ALTER TABLE MC_BLOCK_REFERENCE DROP CONSTRAINT MCBLCKRFERENCEPRNT; ALTER TABLE MC_BLOCK_REFERENCE DROP CONSTRAINT MCBLOCK_REFERENCE0; ALTER TABLE MC_BLOCK_JOB DROP CONSTRAINT MCBLOCKJOBBLOCK_ID; ALTER TABLE MC_ROBOT_BLOCK DROP CONSTRAINT MC_ROBOT_BLOCK_ID; ALTER TABLE MC_ROBOT_BLOCK

Database	Create Tables	Drop Tables
	VARCHAR(255), NAME VARCHAR(255) NOT NULL, ROBOTBYTES BLOB(268435456) NOT NULL, SHA_HASH VARCHAR(255) NOT NULL, VERSION VARCHAR(255) NOT NULL, FOLDER_EX BIGINT NOT NULL, P_ID BIGINT NOT NULL, PRIMARY KEY (ID)); CREATE TABLE MC_TYPE (ID BIGINT NOT NULL, CREATEDBY VARCHAR(255), LASTMODIFIED TIMESTAMP, MODIFIEDBY VARCHAR(255), NAME VARCHAR(255) NOT NULL, SHA_HASH VARCHAR(255) NOT NULL, TYPEBYTES BLOB(268435456) NOT NULL, P_ID BIGINT NOT NULL, PRIMARY KEY (ID)); CREATE TABLE MC_SNIPPET (ID BIGINT NOT NULL, CREATEDBY VARCHAR(255), LASTMODIFIED TIMESTAMP NOT NULL, MODIFIEDBY VARCHAR(255), NAME VARCHAR(255) NOT NULL, SHA_HASH VARCHAR(255) NOT NULL, SNIPPETBYTES BLOB(268435456) NOT NULL, VERSION VARCHAR(255) NOT NULL, P_ID BIGINT NOT NULL, PRIMARY KEY (ID)); CREATE TABLE MC_ROBOSERVER (ID BIGINT NOT NULL, HOST VARCHAR(255) NOT NULL, PORT INTEGER NOT NULL, CLUSTER_ID BIGINT NOT NULL, PRIMARY KEY (ID)); CREATE TABLE MC_SETTINGS (NAME VARCHAR(255) NOT NULL, VALUE VARCHAR(255), PRIMARY KEY (NAME)); CREATE TABLE MC_ROBOT_PARAMETER (ID BIGINT NOT NULL, TYPENAME VARCHAR(255) NOT NULL, VARIABLENAME VARCHAR(255) NOT NULL, ROBOT_ID BIGINT NOT NULL, PRIMARY KEY (ID)); CREATE TABLE MC_ROBOT_REF_TYPE (ID BIGINT NOT NULL, ISINPUT SMALLINT DEFAULT 0 NOT NULL, ISRETURNED SMALLINT DEFAULT 0 NOT NULL, ISSTORED SMALLINT DEFAULT 0 NOT NULL, NAME VARCHAR(255) NOT NULL, ROBOT_ID BIGINT NOT NULL, PRIMARY KEY (ID)); CREATE TABLE MC_ROBOT_REF_SNIPPET (ID BIGINT NOT NULL, NAME VARCHAR(255) NOT NULL, ROBOT_ID BIGINT NOT NULL, PRIMARY KEY (ID)); CREATE TABLE MC_ROBOT_TYPED_VARIABLE (ID BIGINT NOT NULL, TYPENAME VARCHAR(255) NOT NULL, VARIABLENAME VARCHAR(255) NOT NULL, ROBOT_ID BIGINT NOT NULL, PRIMARY KEY (ID)); CREATE TABLE MC_ROBOT_REF_VAR (ID BIGINT NOT NULL, ISRETURNED SMALLINT DEFAULT 0 NOT NULL, ISSTORED SMALLINT DEFAULT 0 NOT NULL, NAME VARCHAR(255), ROBOT_ID BIGINT NOT NULL, PRIMARY KEY (ID)); CREATE TABLE MC_RESOURCE_FILE (ID BIGINT NOT NULL, BYTESIZE INTEGER NOT NULL, BYTES BLOB(268435456) NOT NULL, CREATEDBY VARCHAR(255), LASTMODIFIED TIMESTAMP NOT NULL, MODIFIEDBY VARCHAR(255), NAME VARCHAR(255) NOT NULL, SHA_HASH VARCHAR(255) NOT NULL, FOLDER_EX BIGINT NOT NULL, P_ID BIGINT NOT NULL, PRIMARY KEY (ID)); CREATE TABLE MC_PROJECT (ID BIGINT NOT NULL, ACAO VARCHAR(255), AUTHENTICATE SMALLINT DEFAULT 0, DESCRIPTION VARCHAR(255), NAME	DROP CONSTRAINT MCROBOTBLOCKRBOTID; ALTER TABLE MC_MULTIPLE_ROBOT_BLOCK DROP CONSTRAINT MCMLTPLROBOTBLOCKD; ALTER TABLE MC_CONTAINER_BLOCK DROP CONSTRAINT MCCONTAINERBLOCKID; ALTER TABLE MC_SEQUENTIAL_BLOCK DROP CONSTRAINT MCSEQUENTIALBLOCKD; ALTER TABLE MC_CONCURRENT_BLOCK DROP CONSTRAINT MCCONCURRENTBLOCKD; ALTER TABLE MC_SCRIPT_BLOCK DROP CONSTRAINT MC_SCRIPT_BLOCK_ID; ALTER TABLE MC_FOLDER_ITEM DROP CONSTRAINT MCFOLDERITEMFOLDER; ALTER TABLE MC_ROBOT DROP CONSTRAINT FK_MC_ROBOT_ID; ALTER TABLE MC_ROBOT DROP CONSTRAINT MC_ROBOT_FOLDER_EX; ALTER TABLE MC_ROBOT DROP CONSTRAINT FK_MC_ROBOT_P_ID; ALTER TABLE MC_ROBOT DROP CONSTRAINT UNQ_MC_ROBOT_0; ALTER TABLE MC_TYPE DROP CONSTRAINT FK_MC_TYPE_P_ID; ALTER TABLE MC_TYPE DROP CONSTRAINT FK_MC_TYPE_ID; ALTER TABLE MC_TYPE DROP CONSTRAINT UNQ_MC_TYPE_0; ALTER TABLE MC_SNIPPET DROP CONSTRAINT FK_MC_SNIPPET_P_ID;

Database	Create Tables	Drop Tables
	VARCHAR(255) NOT NULL UNIQUE, CLUSTER_ID BIGINT, PRIMARY KEY (ID)); CREATE TABLE MC_ROBOSERVER_CLUSTER (ID BIGINT NOT NULL, KCU INTEGER, NAME VARCHAR(255) NOT NULL UNIQUE, type SMALLINT DEFAULT 0 NOT NULL, USESSL SMALLINT DEFAULT 0 NOT NULL, SETTINGS_ID BIGINT, PRIMARY KEY (ID)); CREATE TABLE MC_GROUP_TO_ROLE_MAP (ID BIGINT NOT NULL, GROUP_NAME VARCHAR(200) NOT NULL, ROLE_NAME VARCHAR(40) NOT NULL, P_ID BIGINT NOT NULL, PRIMARY KEY (ID)); CREATE TABLE MC_SNIPPET_PARAMETER (ID BIGINT NOT NULL, TYPENAME VARCHAR(255) NOT NULL, VARIABLENAME VARCHAR(255) NOT NULL, SNIPPET_ID BIGINT NOT NULL, PRIMARY KEY (ID)); CREATE TABLE MC_SNIPPET_REF_SNIPPET (ID BIGINT NOT NULL, NAME VARCHAR(255) NOT NULL, SNIPPET_ID BIGINT NOT NULL, PRIMARY KEY (ID)); CREATE TABLE MC_SNIPPET_REF_TYPE (ID BIGINT NOT NULL, ISINPUT SMALLINT DEFAULT 0 NOT NULL, ISRETURNED SMALLINT DEFAULT 0 NOT NULL, ISSTORED SMALLINT DEFAULT 0 NOT NULL, NAME VARCHAR(255), SNIPPET_ID BIGINT NOT NULL, PRIMARY KEY (ID)); CREATE TABLE MC_SNIPPET_TYPED_VAR (ID BIGINT NOT NULL, TYPENAME VARCHAR(255) NOT NULL, VARIABLENAME VARCHAR(255) NOT NULL, SNIPPET_ID BIGINT NOT NULL, PRIMARY KEY (ID)); CREATE TABLE MC_SNIPPET_REF_VAR (ID BIGINT NOT NULL, ISRETURNED SMALLINT DEFAULT 0 NOT NULL, ISSTORED SMALLINT DEFAULT 0 NOT NULL, NAME VARCHAR(255), SNIPPET_ID BIGINT NOT NULL, PRIMARY KEY (ID)); CREATE TABLE MC_OAUTH_APPLICATION (ID BIGINT NOT NULL, CALLBACKURL VARCHAR(255), CONSUMERKEY VARCHAR(255), CONSUMERSECRET CLOB(268435456), NAME VARCHAR(255) NOT NULL, SCOPE VARCHAR(255), SERVICEPROVIDER VARCHAR(255), P_ID BIGINT NOT NULL, PRIMARY KEY (ID)); CREATE TABLE MC_OAUTH_USER (ID BIGINT NOT NULL, ACCESSTOKEN VARCHAR(255), ACCESSTOKENSECRET CLOB(268435456), NAME VARCHAR(255) NOT NULL, REFRESHTOKEN VARCHAR(255), APPLICATION_ID BIGINT NOT NULL, USER_ID VARCHAR(255), PRIMARY KEY (ID)); CREATE TABLE MC_DSLICENSE (TOKEN VARCHAR(255) NOT NULL, EXPIRATION BIGINT, IP VARCHAR(255), USERNAME VARCHAR(255), PRIMARY KEY (TOKEN)); CREATE TABLE MC_USER (USER_NAME VARCHAR(255) NOT NULL, ADMIN_USER SMALLINT DEFAULT 0, CREATION_DATE TIMESTAMP, EMAIL_ADDR VARCHAR(255), FULL_NAME VARCHAR(255), LAST_IP_ADDR VARCHAR(255), LOGIN_TIME TIMESTAMP, LOGIN_COUNT INTEGER NOT NULL, PASS_WORD VARCHAR(255), PRIMARY KEY (USER_NAME)); CREATE TABLE MC_GROUP (ID BIGINT NOT NULL, DESCRIPTION VARCHAR(255), GROUP_NAME VARCHAR(255) NOT NULL, PRIMARY	ALTER TABLE MC_SNIPPET DROP CONSTRAINT FK_MC_SNIPPET_ID; ALTER TABLE MC_SNIPPET DROP CONSTRAINT UNQ_MC_SNIPPET_0; ALTER TABLE MC_ROBOSERVER DROP CONSTRAINT MCRBSERVERCLSTERID; ALTER TABLE MC_ROBOT_PARAMETER DROP CONSTRAINT MCRBTPARAMETERRBTD; ALTER TABLE MC_ROBOT_REF_TYPE DROP CONSTRAINT MCRBOTREFTYPERBTID; ALTER TABLE MC_ROBOT_REF_SNIPPET DROP CONSTRAINT MCRBTRFSNIPPETRBTD; ALTER TABLE MC_ROBOT_TYPED_VARIABLE DROP CONSTRAINT MCRBTTYPDVRBLERBTD; ALTER TABLE MC_ROBOT_REF_VAR DROP CONSTRAINT MCRBOTREFVARRBOTID; ALTER TABLE MC_RESOURCE_FILE DROP CONSTRAINT MC_RESOURCE_FILEID; ALTER TABLE MC_RESOURCE_FILE DROP CONSTRAINT MCRESOURCEFILEP_ID; ALTER TABLE MC_RESOURCE_FILE DROP CONSTRAINT MCRSURCEFILEFLDREX; ALTER TABLE MC_RESOURCE_FILE DROP CONSTRAINT MC_RESOURCE_FILE0; ALTER TABLE MC_PROJECT DROP CONSTRAINT MCPROJECTCLUSTERID; ALTER TABLE MC_ROBOSERVER_CLUSTER DROP CONSTRAINT MCRBSRVRCLSSTTNGSD; ALTER TABLE MC_GROUP_TO_ROLE_MAP

Database	Create Tables	Drop Tables
	KEY (ID)); CREATE TABLE MC_USER_GROUP_REL (ID BIGINT NOT NULL, GROUP_ID BIGINT NOT NULL, USER_ID VARCHAR(255) NOT NULL, PRIMARY KEY (ID)); CREATE TABLE MC_USER_PROP (ID BIGINT NOT NULL, IS_NULL SMALLINT DEFAULT 0, PROP_NAME VARCHAR(255) NOT NULL, PROP_VALUE CLOB(268435456), USER_ID VARCHAR(255) NOT NULL, PRIMARY KEY (ID)); CREATE TABLE MC_TYPE_ATTR (ID BIGINT NOT NULL, ATTR_TYPE VARCHAR(255), COMMENTS CLOB(268435456), DEFAULT_VALUE CLOB(268435456), NAME VARCHAR(255), REQUIRED_ATTR SMALLINT DEFAULT 0, TYPE_ORDER INTEGER, TYPE_ID BIGINT NOT NULL, PRIMARY KEY (ID)); CREATE TABLE MC_RS_CLUSTER_SETTINGS (ID BIGINT NOT NULL, CLUSTER_SETTINGS_ID VARCHAR(255) NOT NULL, EMAIL_LOG_FROM_ADDRESS VARCHAR(255), LOG_INPUT_TO_DB SMALLINT DEFAULT 0, LOG_INPUT_TO_LOG4J SMALLINT DEFAULT 0, LOG_TO_DB SMALLINT DEFAULT 0 NOT NULL, LOG_TO_EMAIL SMALLINT DEFAULT 0 NOT NULL, LOG_TO_LOG4J SMALLINT DEFAULT 0 NOT NULL, MAX_CONCURRENT_ROBOTS INTEGER, MAX_QUEUED_ROBOTS INTEGER, PROF_ENABLED SMALLINT DEFAULT 0, PROF_FILE_OUTPUT_APPEND SMALLINT DEFAULT 0, PROF_FILE_OUTPUT_FILE VARCHAR(255), PROF_LOG_PAGE_URL SMALLINT DEFAULT 0, PROF_OUTPUT_TARGET VARCHAR(20), PROF_TYPE VARCHAR(20), PROF_THRESHOLD INTEGER, EMAIL_LOG_TO_ADDRESS VARCHAR(255), PRIMARY KEY (ID)); CREATE TABLE MC_DATABASE (ID BIGINT NOT NULL, HOST VARCHAR(255) NOT NULL, MAX_ACTIVE_CONNECTIONS INTEGER, MAX_IDLE_CONNECTIONS INTEGER, NAME VARCHAR(255) NOT NULL, PASSWORD VARCHAR(255), SCHEMA_NAME VARCHAR(255) NOT NULL, TYPE VARCHAR(255) NOT NULL, USERNAME VARCHAR(255), CLUSTER_SETTINGS_ID BIGINT NOT NULL, PRIMARY KEY (ID)); CREATE TABLE MC_PROXY_SERVER (ID BIGINT NOT NULL, EXCLUDED_HOST_NAMES VARCHAR(1000), HOST_NAME VARCHAR(255) NOT NULL, PASSWORD VARCHAR(255), PORT_NUMBER INTEGER, USERNAME VARCHAR(255), CLUSTER_SETTINGS_ID BIGINT NOT NULL, PRIMARY KEY (ID)); CREATE TABLE MC_DATABASE_TYPE (ID BIGINT NOT NULL, DRIVER_CLASS VARCHAR(255) NOT NULL, NAME VARCHAR(255) NOT NULL, URL_TEMPLATE VARCHAR(255) NOT NULL, VALIDATION_QUERY VARCHAR(255) NOT NULL, PRIMARY KEY (ID)); CREATE TABLE MC_JAR_FILE (ID BIGINT NOT NULL, BYTES BLOB(268435456), CREATED TIMESTAMP, NAME VARCHAR(255) NOT NULL, FILE_SIZE INTEGER, PRIMARY KEY (ID)); CREATE TABLE APP_KAPPLET (ID VARCHAR(40) NOT NULL, DTYPE VARCHAR(31), CREATED_DATE TIMESTAMP NOT NULL, DESCRIPTION CLOB(268435456), NAME VARCHAR(255) NOT	DROP CONSTRAINT MCGROUPTOROLEMAPPD; ALTER TABLE MC_GROUP_TO_ROLE_MAP DROP CONSTRAINT MCGROUPTOROLE_MAP0; ALTER TABLE MC_SNIPPET_PARAMETER DROP CONSTRAINT MCSNPPTPRMTRSNPPTD; ALTER TABLE MC_SNIPPET_REF_SNIPPET DROP CONSTRAINT MCSNPPTRFSNPSNPPTD; ALTER TABLE MC_SNIPPET_REF_TYPE DROP CONSTRAINT MCSNPPTRFTYPSNPPTD; ALTER TABLE MC_SNIPPET_TYPED_VAR DROP CONSTRAINT MCSNPPTTYPDVSNPPTD; ALTER TABLE MC_SNIPPET_REF_VAR DROP CONSTRAINT MCSNPPTRFVARSNPPTD; ALTER TABLE MC_OAUTH_APPLICATION DROP CONSTRAINT MCUTHAPPLICATIONPD; ALTER TABLE MC_OAUTH_APPLICATION DROP CONSTRAINT MCAUTHAPPLICATION0; ALTER TABLE MC_OAUTH_USER DROP CONSTRAINT MCOAUTHUSERUSER_ID; ALTER TABLE MC_OAUTH_USER DROP CONSTRAINT MCTHUSERPPLCTIONID; ALTER TABLE MC_OAUTH_USER DROP CONSTRAINT MC_OAUTH_USER0; ALTER TABLE MC_GROUP DROP CONSTRAINT UNQ_MC_GROUP_0; ALTER TABLE MC_USER_GROUP_REL DROP CONSTRAINT MCSERGROUPRELSERID; ALTER TABLE MC_USER_GROUP_REL DROP CONSTRAINT MCSERGROUPRELGRPID;

Database	Create Tables	Drop Tables
	NULL, OPTIONS CLOB(268435456), P_ID BIGINT NOT NULL, ICON_ID VARCHAR(255), PRIMARY KEY (ID)); CREATE TABLE APP_MASTER (ID VARCHAR(40) NOT NULL, CREATEDBY VARCHAR(255), ENABLED SMALLINT DEFAULT 0 NOT NULL, FLOW CLOB(268435456), INITIAL_LABEL INTEGER, KAPPLET_TYPE VARCHAR(255) NOT NULL, LAST_INSTALLED TIMESTAMP, LAST_MODIFIED TIMESTAMP NOT NULL, LAST_RUN TIMESTAMP, MODIFIEDBY VARCHAR(255), PRIMARY KEY (ID)); CREATE TABLE APP_INSTALLED (ID VARCHAR(40) NOT NULL, FAVORITE SMALLINT DEFAULT 0, LAST_RUN TIMESTAMP, UPDATED_AT TIMESTAMP, PARENT VARCHAR(40) NOT NULL, USER_ID VARCHAR(255) NOT NULL, PRIMARY KEY (ID)); CREATE TABLE APP_ICON (ID VARCHAR(255) NOT NULL, ICON BLOB(268435456), filename VARCHAR(255), mimeType VARCHAR(255), PRIMARY KEY (ID)); CREATE TABLE APP_UPLOAD (ID VARCHAR(255) NOT NULL, UPLOAD BLOB(268435456) NOT NULL, filename VARCHAR(255) NOT NULL, mimeType VARCHAR(255) NOT NULL, timestamp TIMESTAMP NOT NULL, username VARCHAR(255) NOT NULL, PRIMARY KEY (ID)); CREATE TABLE MC_KAPPLET_PROP (ID BIGINT NOT NULL, IS_NULL SMALLINT DEFAULT 0, PROP_NAME VARCHAR(255) NOT NULL, PROP_VALUE CLOB(268435456), KAPPLET_ID VARCHAR(40) NOT NULL, PRIMARY KEY (ID)); CREATE TABLE APP_SCHEDULE_ERROR (ID BIGINT NOT NULL, ERROR CLOB(268435456), SCHEDULE_ID BIGINT NOT NULL, PRIMARY KEY (ID)); CREATE TABLE APP_SCHEDULE (ID BIGINT NOT NULL, DIRTY SMALLINT DEFAULT 0, LASTRUN TIMESTAMP, NEXTRUN TIMESTAMP, PARAMETERS CLOB(268435456), APP_ID VARCHAR(40) NOT NULL, PRIMARY KEY (ID)); CREATE TABLE APP_SCHEDULE_VALUE (ID VARCHAR(255) NOT NULL, VALUE BLOB(268435456) NOT NULL, SCHEDULE_ID BIGINT NOT NULL, PRIMARY KEY (ID)); CREATE TABLE MC_GLOBAL_KAPPLET_PROP (PROP_NAME VARCHAR(255) NOT NULL, IS_NULL SMALLINT DEFAULT 0, PROP_VALUE CLOB(268435456), PRIMARY KEY (PROP_NAME)); CREATE TABLE APP_ACTION_EXECUTION (ID BIGINT NOT NULL, END_TIME TIMESTAMP, LABEL INTEGER NOT NULL, SEQ_ID INTEGER, START_TIME TIMESTAMP, STATUS VARCHAR(255), TASK_IDS VARCHAR(255), VERSIONLOCK BIGINT, KAPPLET_RUN_ID BIGINT NOT NULL, PRIMARY KEY (ID)); CREATE TABLE APP_DATA_ROW (ID BIGINT NOT NULL, TABLE_ID BIGINT NOT NULL, PRIMARY KEY (ID)); CREATE TABLE APP_DATA_TABLE (ID BIGINT NOT NULL, TABLE_NAME VARCHAR(255) NOT NULL, FLOW_EXECUTION_ID BIGINT NOT NULL, PRIMARY KEY (ID)); CREATE TABLE APP_DATA_STR (ID BIGINT NOT NULL, NAME VARCHAR(255), VALUE VARCHAR(2000) NOT	ALTER TABLE MC_USER_GROUP_REL DROP CONSTRAINT MC_USER_GROUP_REL0; ALTER TABLE MC_USER_PROP DROP CONSTRAINT MCUSER_PROPUSER_ID; ALTER TABLE MC_USER_PROP DROP CONSTRAINT UNQ_MC_USER_PROP_0; ALTER TABLE MC_TYPE_ATTR DROP CONSTRAINT MCTYPE_ATTRTYPE_ID; ALTER TABLE MC_DATABASE DROP CONSTRAINT MCDTBSCLSTRSTTNGSD; ALTER TABLE MC_PROXY_SERVER DROP CONSTRAINT MCPRXYCLSTRSTTNGSD; ALTER TABLE APP_KAPPLET DROP CONSTRAINT APP_KAPPLET_P_ID; ALTER TABLE APP_KAPPLET DROP CONSTRAINT APP_KAPPLETICON_ID; ALTER TABLE APP_MASTER DROP CONSTRAINT FK_APP_MASTER_ID; ALTER TABLE APP_INSTALLED DROP CONSTRAINT APPINSTALLEDPARENT; ALTER TABLE APP_INSTALLED DROP CONSTRAINT APP_INSTALLED_ID; ALTER TABLE APP_INSTALLED DROP CONSTRAINT APPINSTALLEDUSERID; ALTER TABLE MC_KAPPLET_PROP DROP CONSTRAINT MCKPPLTPROPKPPLTID; ALTER TABLE MC_KAPPLET_PROP DROP CONSTRAINT MC_KAPPLET_PROP0; ALTER TABLE APP_SCHEDULE_ERROR

Database	Create Tables	Drop Tables
	NULL, RESULT_ID BIGINT NOT NULL, PRIMARY KEY (ID)); CREATE TABLE APP_DATA_LOB (ID BIGINT NOT NULL, LARGE SMALLINT DEFAULT 0, NAME VARCHAR(255), VALUE CLOB(268435456), RESULT_ID BIGINT NOT NULL, PRIMARY KEY (ID)); CREATE TABLE APP_FLOW_ERROR (ID BIGINT NOT NULL, ERROR CLOB(268435456), EXECUTION_ID BIGINT NOT NULL, PRIMARY KEY (ID)); CREATE TABLE APP_RUN (ID BIGINT NOT NULL, ABORTED SMALLINT DEFAULT 0, FLOW CLOB(268435456), INITIAL_LABEL INTEGER NOT NULL, LASTEXECUTIONPATHCHANGE BIGINT, OPTIONS CLOB(268435456), PARAMETERS CLOB(268435456), SIGNATURES CLOB(268435456), VERSIONLOCK BIGINT, KAPPLET_ID VARCHAR(40) NOT NULL, PRIMARY KEY (ID)); CREATE TABLE APP_RUN_VALUE (ID VARCHAR(255) NOT NULL, VALUE BLOB(268435456) NOT NULL, RUN_ID BIGINT NOT NULL, PRIMARY KEY (ID)); CREATE TABLE MC_FOLDER (ID BIGINT NOT NULL, DTYPE VARCHAR(31), PATH VARCHAR(255), PRIMARY KEY (ID)); CREATE TABLE MC_ROOT_FOLDER (ID BIGINT NOT NULL, P_ID BIGINT NOT NULL, PRIMARY KEY (ID)); CREATE TABLE MC_SUBFOLDER (ID BIGINT NOT NULL, NAME VARCHAR(255) NOT NULL, PARENT_ID BIGINT NOT NULL, PRIMARY KEY (ID)); CREATE TABLE APP_RUN_APP_SCHEDULE (schedules_ID BIGINT NOT NULL, kappletRuns_ID BIGINT NOT NULL, PRIMARY KEY (schedules_ID, kappletRuns_ID)); ALTER TABLE MC_SCHEDULE ADD CONSTRAINT UNQ_MC_SCHEDULE_0 UNIQUE (P_ID, NAME); ALTER TABLE MC_BLOCK ADD CONSTRAINT UNQ_MC_BLOCK_0 UNIQUE (P_ID, NAME); ALTER TABLE MC_BLOCK_REFERENCE ADD CONSTRAINT MCBLOCK_REFERENCE0 UNIQUE (PARENT, NAME); ALTER TABLE MC_ROBOT ADD CONSTRAINT UNQ_MC_ROBOT_0 UNIQUE (P_ID, NAME, FOLDER_EX); ALTER TABLE MC_TYPE ADD CONSTRAINT UNQ_MC_TYPE_0 UNIQUE (P_ID, NAME); ALTER TABLE MC_SNIPPET ADD CONSTRAINT UNQ_MC_SNIPPET_0 UNIQUE (P_ID, NAME); ALTER TABLE MC_RESOURCE_FILE ADD CONSTRAINT MC_RESOURCE_FILE0 UNIQUE (P_ID, NAME, FOLDER_EX); ALTER TABLE MC_GROUP_TO_ROLE_MAP ADD CONSTRAINT MCGROUPTOROLE_MAP0 UNIQUE (P_ID, ROLE_NAME, GROUP_NAME); ALTER TABLE MC_OAUTH_APPLICATION ADD CONSTRAINT MCAUTHAPPLICATION0 UNIQUE (P_ID, NAME); ALTER TABLE MC_OAUTH_USER ADD CONSTRAINT MC_OAUTH_USER0 UNIQUE (APPLICATION_ID, NAME); ALTER TABLE MC_GROUP ADD CONSTRAINT UNQ_MC_GROUP_0 UNIQUE (GROUP_NAME); ALTER TABLE MC_USER_GROUP_REL ADD CONSTRAINT MC_USER_GROUP_REL0 UNIQUE (USER_ID, GROUP_ID); ALTER TABLE MC_USER_PROP ADD CONSTRAINT UNQ_MC_USER_PROP_0 UNIQUE (ID, PROP_NAME); ALTER TABLE MC_KAPPLET_PROP	DROP CONSTRAINT PPSCHDLERRORSCHDLD; ALTER TABLE APP_SCHEDULE DROP CONSTRAINT APP_SCHEDULEAPP_ID; ALTER TABLE APP_SCHEDULE_VALUE DROP CONSTRAINT PPSCHDLVALUESCHDLD; ALTER TABLE APP_ACTION_EXECUTION DROP CONSTRAINT PPCTNXCTONKPPLTRND; ALTER TABLE APP_ACTION_EXECUTION DROP CONSTRAINT PPACTIONEXECUTION0; ALTER TABLE APP_DATA_ROW DROP CONSTRAINT APPDATAROWTABLE_ID; ALTER TABLE APP_DATA_TABLE DROP CONSTRAINT PPDTTBLEFLWXCTONID; ALTER TABLE APP_DATA_TABLE DROP CONSTRAINT APP_DATA_TABLE0; ALTER TABLE APP_DATA_STR DROP CONSTRAINT APPDATASTRRESULTID; ALTER TABLE APP_DATA_LOB DROP CONSTRAINT APPDATALOBRESULTID; ALTER TABLE APP_FLOW_ERROR DROP CONSTRAINT PPFLWERRORXCTIONID; ALTER TABLE APP_RUN DROP CONSTRAINT APP_RUN_KAPPLET_ID; ALTER TABLE APP_RUN_VALUE DROP CONSTRAINT APPRUN_VALUERUN_ID; ALTER TABLE MC_ROOT_FOLDER DROP CONSTRAINT MC_ROOT_FOLDERP_ID; ALTER TABLE MC_ROOT_FOLDER DROP CONSTRAINT MC_ROOT_FOLDER_ID;

Database	Create Tables	Drop Tables
	ADD CONSTRAINT MC_KAPPLET_PROP0 UNIQUE (ID, PROP_NAME); ALTER TABLE APP_ACTION_EXECUTION ADD CONSTRAINT PPACTIONEXECUTION0 UNIQUE (KAPPLET_RUN_ID, LABEL); ALTER TABLE APP_DATA_TABLE ADD CONSTRAINT APP_DATA_TABLE0 UNIQUE (FLOW_EXECUTION_ID, TABLE_NAME); ALTER TABLE MC_SUBFOLDER ADD CONSTRAINT UNQ_MC_SUBFOLDER_0 UNIQUE (NAME, PARENT_ID); ALTER TABLE MC_SCHEDULE ADD CONSTRAINT MCSCHDULEBLCKJOBID FOREIGN KEY (BLOCKJOB_ID) REFERENCES MC_BLOCK_JOB (ID); ALTER TABLE MC_SCHEDULE ADD CONSTRAINT MC_SCHEDULE_P_ID FOREIGN KEY (P_ID) REFERENCES MC_PROJECT (ID); ALTER TABLE MC_SCHEDULE ADD CONSTRAINT MCSCHEDULECLSTERID FOREIGN KEY (CLUSTER_ID) REFERENCES MC_ROBOSERVER_CLUSTER (ID); ALTER TABLE MC_BLOCK ADD CONSTRAINT FK_MC_BLOCK_P_ID FOREIGN KEY (P_ID) REFERENCES MC_PROJECT (ID); ALTER TABLE MC_BLOCK_REFERENCE ADD CONSTRAINT MCBLCKRFERENCECHLD FOREIGN KEY (CHILD) REFERENCES MC_BLOCK (ID); ALTER TABLE MC_BLOCK_REFERENCE ADD CONSTRAINT MCBLCKRFERENCEPRNT FOREIGN KEY (PARENT) REFERENCES MC_BLOCK (ID); ALTER TABLE MC_BLOCK_JOB ADD CONSTRAINT MCBLOCKJOBBLOCK_ID FOREIGN KEY (BLOCK_ID) REFERENCES MC_BLOCK (ID); ALTER TABLE MC_ROBOT_BLOCK ADD CONSTRAINT MC_ROBOT_BLOCK_ID FOREIGN KEY (ID) REFERENCES MC_BLOCK (ID); ALTER TABLE MC_ROBOT_BLOCK ADD CONSTRAINT MCROBOTBLOCKRBOTID FOREIGN KEY (ROBOT_ID) REFERENCES MC_FOLDER_ITEM (ID); ALTER TABLE MC_MULTIPLE_ROBOT_BLOCK ADD CONSTRAINT MCMLTPLROBOTBLOCKD FOREIGN KEY (ID) REFERENCES MC_BLOCK (ID); ALTER TABLE MC_CONTAINER_BLOCK ADD CONSTRAINT MCCONTAINERBLOCKID FOREIGN KEY (ID) REFERENCES MC_BLOCK (ID); ALTER TABLE MC_SEQUENTIAL_BLOCK ADD CONSTRAINT MCSEQUENTIALBLOCKD FOREIGN KEY (ID) REFERENCES MC_BLOCK (ID); ALTER TABLE MC_CONCURRENT_BLOCK ADD CONSTRAINT MCCONCURRENTBLOCKD FOREIGN KEY (ID) REFERENCES MC_BLOCK (ID); ALTER TABLE MC_SCRIPT_BLOCK ADD CONSTRAINT MC_SCRIPT_BLOCK_ID FOREIGN KEY (ID) REFERENCES MC_BLOCK (ID); ALTER TABLE MC_FOLDER_ITEM ADD CONSTRAINT MCFOLDERITEMFOLDER FOREIGN KEY (FOLDER) REFERENCES MC_FOLDER (ID); ALTER TABLE MC_ROBOT ADD CONSTRAINT FK_MC_ROBOT_ID FOREIGN KEY (ID) REFERENCES MC_FOLDER_ITEM (ID); ALTER TABLE MC_ROBOT ADD CONSTRAINT MC_ROBOT_FOLDER_EX FOREIGN KEY (FOLDER_EX) REFERENCES MC_FOLDER (ID); ALTER TABLE	ALTER TABLE MC_SUBFOLDER DROP CONSTRAINT MCSUBFOLDERPRENTID; ALTER TABLE MC_SUBFOLDER DROP CONSTRAINT FK_MC_SUBFOLDER_ID; ALTER TABLE MC_SUBFOLDER DROP CONSTRAINT UNQ_MC_SUBFOLDER_0; ALTER TABLE APP_RUN_APP_SCHEDULE DROP CONSTRAINT PPRNPPSCHDLschdlsD; ALTER TABLE APP_RUN_APP_SCHEDULE DROP CONSTRAINT PPRNPPSCHkppltRnsD; DROP TABLE MC_SCHEDULE; DROP TABLE MC_BLOCK; DROP TABLE MC_BLOCK_REFERENCE; DROP TABLE MC_BLOCK_JOB; DROP TABLE MC_ROBOT_BLOCK; DROP TABLE MC_MULTIPLE_ROBOT_BLOCK; DROP TABLE MC_CONTAINER_BLOCK; DROP TABLE MC_SEQUENTIAL_BLOCK; DROP TABLE MC_CONCURRENT_BLOCK; DROP TABLE MC_SCRIPT_BLOCK; DROP TABLE MC_FOLDER_ITEM; DROP TABLE MC_ROBOT; DROP TABLE MC_TYPE; DROP TABLE MC_SNIPPET; DROP TABLE MC_ROBOSERVER; DROP TABLE MC_SETTINGS; DROP TABLE MC_ROBOT_PARAMETER; DROP TABLE MC_ROBOT_REF_TYPE; DROP TABLE MC_ROBOT_REF_SNIPPET; DROP TABLE MC_ROBOT_TYPED_VARIABLE;

Database	Create Tables	Drop Tables
	MC_ROBOT ADD CONSTRAINT FK_MC_ROBOT_P_ID FOREIGN KEY (P_ID) REFERENCES MC_PROJECT (ID); ALTER TABLE MC_TYPE ADD CONSTRAINT FK_MC_TYPE_P_ID FOREIGN KEY (P_ID) REFERENCES MC_PROJECT (ID); ALTER TABLE MC_TYPE ADD CONSTRAINT FK_MC_TYPE_ID FOREIGN KEY (ID) REFERENCES MC_FOLDER_ITEM (ID); ALTER TABLE MC_SNIPPET ADD CONSTRAINT FK_MC_SNIPPET_P_ID FOREIGN KEY (P_ID) REFERENCES MC_PROJECT (ID); ALTER TABLE MC_SNIPPET ADD CONSTRAINT FK_MC_SNIPPET_ID FOREIGN KEY (ID) REFERENCES MC_FOLDER_ITEM (ID); ALTER TABLE MC_ROBOSERVER ADD CONSTRAINT MCRBSERVERCLSTERID FOREIGN KEY (CLUSTER_ID) REFERENCES MC_ROBOSERVER_CLUSTER (ID); ALTER TABLE MC_ROBOT_PARAMETER ADD CONSTRAINT MCRBTPARAMETERRBTD FOREIGN KEY (ROBOT_ID) REFERENCES MC_FOLDER_ITEM (ID); ALTER TABLE MC_ROBOT_REF_TYPE ADD CONSTRAINT MCRBOTREFTYPERBTID FOREIGN KEY (ROBOT_ID) REFERENCES MC_FOLDER_ITEM (ID); ALTER TABLE MC_ROBOT_REF_SNIPPET ADD CONSTRAINT MCRBTRFSNIPPETRBTD FOREIGN KEY (ROBOT_ID) REFERENCES MC_FOLDER_ITEM (ID); ALTER TABLE MC_ROBOT_TYPED_VARIABLE ADD CONSTRAINT MCRBTTYPDVRBLERBTD FOREIGN KEY (ROBOT_ID) REFERENCES MC_FOLDER_ITEM (ID); ALTER TABLE MC_ROBOT_REF_VAR ADD CONSTRAINT MCRBOTREFVARRBOTID FOREIGN KEY (ROBOT_ID) REFERENCES MC_FOLDER_ITEM (ID); ALTER TABLE MC_RESOURCE_FILE ADD CONSTRAINT MC_RESOURCE_FILEID FOREIGN KEY (ID) REFERENCES MC_FOLDER_ITEM (ID); ALTER TABLE MC_RESOURCE_FILE ADD CONSTRAINT MCRESOURCEFILEP_ID FOREIGN KEY (P_ID) REFERENCES MC_PROJECT (ID); ALTER TABLE MC_RESOURCE_FILE ADD CONSTRAINT MCRSURCEFILEFLDREX FOREIGN KEY (FOLDER_EX) REFERENCES MC_FOLDER (ID); ALTER TABLE MC_PROJECT ADD CONSTRAINT MCPROJECTCLUSTERID FOREIGN KEY (CLUSTER_ID) REFERENCES MC_ROBOSERVER_CLUSTER (ID); ALTER TABLE MC_ROBOSERVER_CLUSTER ADD CONSTRAINT MCRBSRVRCLSSTTNGSD FOREIGN KEY (SETTINGS_ID) REFERENCES MC_RS_CLUSTER_SETTINGS (ID); ALTER TABLE MC_GROUP_TO_ROLE_MAP ADD CONSTRAINT MCGROUPTOROLEMAPPD FOREIGN KEY (P_ID) REFERENCES MC_PROJECT (ID); ALTER TABLE MC_SNIPPET_PARAMETER ADD CONSTRAINT MCSNPPTPRMTRSNPPTD FOREIGN KEY (SNIPPET_ID) REFERENCES MC_FOLDER_ITEM (ID); ALTER TABLE MC_SNIPPET_REF_SNIPPET ADD CONSTRAINT MCSNPPTRFSNPSNPPTD FOREIGN KEY (SNIPPET_ID) REFERENCES MC_FOLDER_ITEM (ID); ALTER TABLE MC_SNIPPET_REF_TYPE ADD CONSTRAINT MCSNPPTRFTYPSNPPTD FOREIGN KEY (SNIPPET_ID) REFERENCES MC_FOLDER_ITEM	DROP TABLE MC_ROBOT_REF_VAR; DROP TABLE MC_RESOURCE_FILE; DROP TABLE MC_PROJECT; DROP TABLE MC_ROBOSERVER_CLUSTER; DROP TABLE MC_GROUP_TO_ROLE_MAP; DROP TABLE MC_SNIPPET_PARAMETER; DROP TABLE MC_SNIPPET_REF_SNIPPET; DROP TABLE MC_SNIPPET_REF_TYPE; DROP TABLE MC_SNIPPET_TYPED_VAR; DROP TABLE MC_SNIPPET_REF_VAR; DROP TABLE MC_OAUTH_APPLICATION; DROP TABLE MC_OAUTH_USER; DROP TABLE MC_DSLICENSE; DROP TABLE MC_USER; DROP TABLE MC_GROUP; DROP TABLE MC_USER_GROUP_REL; DROP TABLE MC_USER_PROP; DROP TABLE MC_TYPE_ATTR; DROP TABLE MC_RS_CLUSTER_SETTINGS; DROP TABLE MC_DATABASE; DROP TABLE MC_PROXY_SERVER; DROP TABLE MC_DATABASE_TYPE; DROP TABLE MC_JAR_FILE; DROP TABLE APP_KAPPLET; DROP TABLE APP_MASTER; DROP TABLE APP_INSTALLED; DROP TABLE APP_ICON; DROP TABLE APP_UPLOAD; DROP TABLE MC_KAPPLET_PROP; DROP TABLE APP_SCHEDULE_ERROR; DROP TABLE

Database	Create Tables	Drop Tables
	(ID); ALTER TABLE MC_SNIPPET_TYPED_VAR ADD CONSTRAINT MCSNPPTTYPDVSNPPTD FOREIGN KEY (SNIPPET_ID) REFERENCES MC_FOLDER_ITEM (ID); ALTER TABLE MC_SNIPPET_REF_VAR ADD CONSTRAINT MCSNPPTRFVARSNPPTD FOREIGN KEY (SNIPPET_ID) REFERENCES MC_FOLDER_ITEM (ID); ALTER TABLE MC_OAUTH_APPLICATION ADD CONSTRAINT MCUTHAPPLICATIONPD FOREIGN KEY (P_ID) REFERENCES MC_PROJECT (ID); ALTER TABLE MC_OAUTH_USER ADD CONSTRAINT MCOAUTHUSERUSER_ID FOREIGN KEY (USER_ID) REFERENCES MC_USER (USER_NAME); ALTER TABLE MC_OAUTH_USER ADD CONSTRAINT MCTHUSERPPLCTIONID FOREIGN KEY (APPLICATION_ID) REFERENCES MC_OAUTH_APPLICATION (ID); ALTER TABLE MC_USER_GROUP_REL ADD CONSTRAINT MCSERGROUPRELSERID FOREIGN KEY (USER_ID) REFERENCES MC_USER (USER_NAME); ALTER TABLE MC_USER_GROUP_REL ADD CONSTRAINT MCSERGROUPRELGRPID FOREIGN KEY (GROUP_ID) REFERENCES MC_GROUP (ID); ALTER TABLE MC_USER_PROP ADD CONSTRAINT MCUSER_PROPUSER_ID FOREIGN KEY (USER_ID) REFERENCES MC_USER (USER_NAME); ALTER TABLE MC_TYPE_ATTR ADD CONSTRAINT MCTYPE_ATTRTYPE_ID FOREIGN KEY (TYPE_ID) REFERENCES MC_FOLDER_ITEM (ID); ALTER TABLE MC_DATABASE ADD CONSTRAINT MCDTBSCLSTRSTTNGSD FOREIGN KEY (CLUSTER_SETTINGS_ID) REFERENCES MC_RS_CLUSTER_SETTINGS (ID); ALTER TABLE MC_PROXY_SERVER ADD CONSTRAINT MCPRXYCLSTRSTTNGSD FOREIGN KEY (CLUSTER_SETTINGS_ID) REFERENCES MC_RS_CLUSTER_SETTINGS (ID); ALTER TABLE APP_KAPPLET ADD CONSTRAINT APP_KAPPLET_P_ID FOREIGN KEY (P_ID) REFERENCES MC_PROJECT (ID); ALTER TABLE APP_KAPPLET ADD CONSTRAINT APP_KAPPLETICON_ID FOREIGN KEY (ICON_ID) REFERENCES APP_ICON (ID); ALTER TABLE APP_MASTER ADD CONSTRAINT FK_APP_MASTER_ID FOREIGN KEY (ID) REFERENCES APP_KAPPLET (ID); ALTER TABLE APP_INSTALLED ADD CONSTRAINT APPINSTALLEDPARENT FOREIGN KEY (PARENT) REFERENCES APP_KAPPLET (ID); ALTER TABLE APP_INSTALLED ADD CONSTRAINT APP_INSTALLED_ID FOREIGN KEY (ID) REFERENCES APP_KAPPLET (ID); ALTER TABLE APP_INSTALLED ADD CONSTRAINT APPINSTALLEDUSERID FOREIGN KEY (USER_ID) REFERENCES MC_USER (USER_NAME); ALTER TABLE MC_KAPPLET_PROP ADD CONSTRAINT MCKPPLTPROPKPPLTID FOREIGN KEY (KAPPLET_ID) REFERENCES APP_KAPPLET (ID); ALTER TABLE APP_SCHEDULE_ERROR ADD CONSTRAINT PPSCHDLERRORSCHDLD FOREIGN KEY (SCHEDULE_ID) REFERENCES APP_SCHEDULE (ID); ALTER TABLE APP_SCHEDULE ADD	APP_SCHEDULE; DROP TABLE APP_SCHEDULE_VALUE; DROP TABLE MC_GLOBAL_KAPPLET_PROP; DROP TABLE APP_ACTION_EXECUTION; DROP TABLE APP_DATA_ROW; DROP TABLE APP_DATA_TABLE; DROP TABLE APP_DATA_STR; DROP TABLE APP_DATA_LOB; DROP TABLE APP_FLOW_ERROR; DROP TABLE APP_RUN; DROP TABLE APP_RUN_VALUE; DROP TABLE MC_FOLDER; DROP TABLE MC_ROOT_FOLDER; DROP TABLE MC_SUBFOLDER; DROP TABLE APP_RUN_APP_SCHEDULE; DROP SEQUENCE SEQ_GEN_SEQUENCE; DROP TABLE APP_SEQUENCES;

Database	Create Tables	Drop Tables
	CONSTRAINT APP_SCHEDULEAPP_ID FOREIGN KEY (APP_ID) REFERENCES APP_KAPPLET (ID); ALTER TABLE APP_SCHEDULE_VALUE ADD CONSTRAINT PPSCHDLVALUESCHDLD FOREIGN KEY (SCHEDULE_ID) REFERENCES APP_SCHEDULE (ID); ALTER TABLE APP_ACTION_EXECUTION ADD CONSTRAINT PPCTNXCTONKPPLTRND FOREIGN KEY (KAPPLET_RUN_ID) REFERENCES APP_RUN (ID); ALTER TABLE APP_DATA_ROW ADD CONSTRAINT APPDATAROWTABLE_ID FOREIGN KEY (TABLE_ID) REFERENCES APP_DATA_TABLE (ID); ALTER TABLE APP_DATA_TABLE ADD CONSTRAINT PPDTTBLEFLWXCTONID FOREIGN KEY (FLOW_EXECUTION_ID) REFERENCES APP_ACTION_EXECUTION (ID); ALTER TABLE APP_DATA_STR ADD CONSTRAINT APPDATASTRRESULTID FOREIGN KEY (RESULT_ID) REFERENCES APP_DATA_ROW (ID); ALTER TABLE APP_DATA_LOB ADD CONSTRAINT APPDATALOBRESULTID FOREIGN KEY (RESULT_ID) REFERENCES APP_DATA_ROW (ID); ALTER TABLE APP_FLOW_ERROR ADD CONSTRAINT PPFLWERRORXCTIONID FOREIGN KEY (EXECUTION_ID) REFERENCES APP_ACTION_EXECUTION (ID); ALTER TABLE APP_RUN ADD CONSTRAINT APP_RUN_KAPPLET_ID FOREIGN KEY (KAPPLET_ID) REFERENCES APP_KAPPLET (ID); ALTER TABLE APP_RUN_VALUE ADD CONSTRAINT APPRUN_VALUERUN_ID FOREIGN KEY (RUN_ID) REFERENCES APP_RUN (ID); ALTER TABLE MC_ROOT_FOLDER ADD CONSTRAINT MC_ROOT_FOLDERP_ID FOREIGN KEY (P_ID) REFERENCES MC_PROJECT (ID); ALTER TABLE MC_ROOT_FOLDER ADD CONSTRAINT MC_ROOT_FOLDER_ID FOREIGN KEY (ID) REFERENCES MC_FOLDER (ID); ALTER TABLE MC_SUBFOLDER ADD CONSTRAINT MCSUBFOLDERPRENTID FOREIGN KEY (PARENT_ID) REFERENCES MC_FOLDER (ID); ALTER TABLE MC_SUBFOLDER ADD CONSTRAINT FK_MC_SUBFOLDER_ID FOREIGN KEY (ID) REFERENCES MC_FOLDER (ID); ALTER TABLE APP_RUN_APP_SCHEDULE ADD CONSTRAINT PPRNPPSCHDLschdlsD FOREIGN KEY (schedules_ID) REFERENCES APP_SCHEDULE (ID); ALTER TABLE APP_RUN_APP_SCHEDULE ADD CONSTRAINT PPRNPPSCHkppltRnsD FOREIGN KEY (kappletRuns_ID) REFERENCES APP_RUN (ID); CREATE TABLE APP_SEQUENCES (TABLE_NAME VARCHAR(50) NOT NULL, NEXT_ID DECIMAL(15), PRIMARY KEY (TABLE_NAME)); INSERT INTO APP_SEQUENCES(TABLE_NAME, NEXT_ID) values ('DATA_TABLE', 0); INSERT INTO APP_SEQUENCES(TABLE_NAME, NEXT_ID) values ('DATA_LOB', 0); CREATE SEQUENCE SEQ_GEN_SEQUENCE INCREMENT BY 50 START WITH 50; INSERT INTO APP_SEQUENCES(TABLE_NAME, NEXT_ID) values ('DATA_ROW', 0); INSERT	

Database	Create Tables	Drop Tables
	INTO APP_SEQUENCES(TABLE_NAME, NEXT_ID) values ('DATA_STR', 0); INSERT INTO APP_SEQUENCES(TABLE_NAME, NEXT_ID) values ('FLOW_ACTION_EXECUTION', 0);	
Derby	CREATE TABLE MC_SCHEDULE (ID BIGINT NOT NULL, BAD_INPUT SMALLINT DEFAULT 0, ACTIVE SMALLINT DEFAULT 0, CREATEDBY VARCHAR(255), DIRTY SMALLINT DEFAULT 0, EMAILS VARCHAR(255), MAXOBJECTSEXTRACTED INTEGER, MAXRUNTIME INTEGER, MODIFIEDBY VARCHAR(255), NAME VARCHAR(255) NOT NULL, NEXTRUN TIMESTAMP, PREVIOUSRUN TIMESTAMP, TOTALRUNS INTEGER, USEEMAILNOTIFICATION SMALLINT DEFAULT 0, CLUSTER_ID BIGINT, P_ID BIGINT NOT NULL, BLOCKJOB_ID BIGINT, PRIMARY KEY (ID)); CREATE TABLE MC_BLOCK (ID BIGINT NOT NULL, DTYPE VARCHAR(31), NAME VARCHAR(255) NOT NULL, NAMEDBLOCK SMALLINT DEFAULT 0, P_ID BIGINT NOT NULL, PRIMARY KEY (ID)); CREATE TABLE MC_BLOCK_REFERENCE (ID BIGINT NOT NULL, ENABLED SMALLINT DEFAULT 0 NOT NULL, ORDER_VAL INTEGER NOT NULL, NAME VARCHAR(255) NOT NULL, CHILD BIGINT, PARENT BIGINT NOT NULL, PRIMARY KEY (ID)); CREATE TABLE MC_BLOCK_JOB (ID BIGINT NOT NULL, ACTIVE SMALLINT DEFAULT 0 NOT NULL, BLOCKINPUT CLOB(2147483647), DISPLAY_NAMES CLOB(2147483647) NOT NULL, BLOCK_ID BIGINT NOT NULL, PRIMARY KEY (ID)); CREATE TABLE MC_ROBOT_BLOCK (ID BIGINT NOT NULL, ROBOT_ID BIGINT, PRIMARY KEY (ID)); CREATE TABLE MC_MULTIPLE_ROBOT_BLOCK (ID BIGINT NOT NULL, DISPLAY_NAME VARCHAR(255), STRATEGY VARCHAR(255), VALUE VARCHAR(255), PRIMARY KEY (ID)); CREATE TABLE MC_CONTAINER_BLOCK (ID BIGINT NOT NULL, MAPPING BLOB(2147483647), PRIMARY KEY (ID)); CREATE TABLE MC_SEQUENTIAL_BLOCK (ID BIGINT NOT NULL, PRIMARY KEY (ID)); CREATE TABLE MC_CONCURRENT_BLOCK (ID BIGINT NOT NULL, PRIMARY KEY (ID)); CREATE TABLE MC_SCRIPT_BLOCK (ID BIGINT NOT NULL, SCRIPT VARCHAR(255), PRIMARY KEY (ID)); CREATE TABLE MC_FOLDER_ITEM (ID BIGINT NOT NULL, DTYPE VARCHAR(31), FOLDER BIGINT NOT NULL, PRIMARY KEY (ID)); CREATE TABLE MC_ROBOT (ID BIGINT NOT NULL, BYTESIZE INTEGER NOT NULL, CONTAINSRETURNOBJECT SMALLINT DEFAULT 0, CREATEDBY VARCHAR(255), LASTMODIFIED TIMESTAMP NOT NULL, MODIFIEDBY VARCHAR(255), NAME VARCHAR(255) NOT NULL, ROBOTBYTES BLOB(2147483647) NOT NULL, SHA_HASH VARCHAR(255) NOT NULL, VERSION VARCHAR(255) NOT NULL, FOLDER_EX BIGINT NOT NULL, P_ID BIGINT NOT NULL, PRIMARY KEY (ID)); CREATE TABLE MC_TYPE (ID BIGINT NOT NULL, CREATEDBY VARCHAR(255),	ALTER TABLE MC_SCHEDULE DROP CONSTRAINT MCSCHDULEBLCKJOBID; ALTER TABLE MC_SCHEDULE DROP CONSTRAINT MC_SCHEDULE_P_ID; ALTER TABLE MC_SCHEDULE DROP CONSTRAINT MCSCHEDULECLSTERID; ALTER TABLE MC_SCHEDULE DROP CONSTRAINT UNQ_MC_SCHEDULE_0; ALTER TABLE MC_BLOCK DROP CONSTRAINT FK_MC_BLOCK_P_ID; ALTER TABLE MC_BLOCK DROP CONSTRAINT UNQ_MC_BLOCK_0; ALTER TABLE MC_BLOCK_REFERENCE DROP CONSTRAINT MCBLCKRFERENCECHLD; ALTER TABLE MC_BLOCK_REFERENCE DROP CONSTRAINT MCBLCKRFERENCEPRNT; ALTER TABLE MC_BLOCK_REFERENCE DROP CONSTRAINT MCBLOCK_REFERENCE0; ALTER TABLE MC_BLOCK_JOB DROP CONSTRAINT MCBLOCKJOBBLOCK_ID; ALTER TABLE MC_ROBOT_BLOCK DROP CONSTRAINT MC_ROBOT_BLOCK_ID; ALTER TABLE MC_ROBOT_BLOCK DROP CONSTRAINT MCROBOTBLOCKRBOTID; ALTER TABLE MC_MULTIPLE_ROBOT_BLOCK DROP CONSTRAINT MCMLTPLROBOTBLOCKD; ALTER TABLE

Database	Create Tables	Drop Tables
	LASTMODIFIED TIMESTAMP, MODIFIEDBY VARCHAR(255), NAME VARCHAR(255) NOT NULL, SHA_HASH VARCHAR(255) NOT NULL, TYPEBYTES BLOB(2147483647) NOT NULL, P_ID BIGINT NOT NULL, PRIMARY KEY (ID)); CREATE TABLE MC_SNIPPET (ID BIGINT NOT NULL, CREATEDBY VARCHAR(255), LASTMODIFIED TIMESTAMP NOT NULL, MODIFIEDBY VARCHAR(255), NAME VARCHAR(255) NOT NULL, SHA_HASH VARCHAR(255) NOT NULL, SNIPPETBYTES BLOB(2147483647) NOT NULL, VERSION VARCHAR(255) NOT NULL, P_ID BIGINT NOT NULL, PRIMARY KEY (ID)); CREATE TABLE MC_ROBOSERVER (ID BIGINT NOT NULL, HOST VARCHAR(255) NOT NULL, PORT INTEGER NOT NULL, CLUSTER_ID BIGINT NOT NULL, PRIMARY KEY (ID)); CREATE TABLE MC_SETTINGS (NAME VARCHAR(255) NOT NULL, VALUE VARCHAR(255), PRIMARY KEY (NAME)); CREATE TABLE MC_ROBOT_PARAMETER (ID BIGINT NOT NULL, TYPENAME VARCHAR(255) NOT NULL, VARIABLENAME VARCHAR(255) NOT NULL, ROBOT_ID BIGINT NOT NULL, PRIMARY KEY (ID)); CREATE TABLE MC_ROBOT_REF_TYPE (ID BIGINT NOT NULL, ISINPUT SMALLINT DEFAULT 0 NOT NULL, ISRETURNED SMALLINT DEFAULT 0 NOT NULL, ISSTORED SMALLINT DEFAULT 0 NOT NULL, NAME VARCHAR(255) NOT NULL, ROBOT_ID BIGINT NOT NULL, PRIMARY KEY (ID)); CREATE TABLE MC_ROBOT_REF_SNIPPET (ID BIGINT NOT NULL, NAME VARCHAR(255) NOT NULL, ROBOT_ID BIGINT NOT NULL, PRIMARY KEY (ID)); CREATE TABLE MC_ROBOT_TYPED_VARIABLE (ID BIGINT NOT NULL, TYPENAME VARCHAR(255) NOT NULL, VARIABLENAME VARCHAR(255) NOT NULL, ROBOT_ID BIGINT NOT NULL, PRIMARY KEY (ID)); CREATE TABLE MC_ROBOT_REF_VAR (ID BIGINT NOT NULL, ISRETURNED SMALLINT DEFAULT 0 NOT NULL, ISSTORED SMALLINT DEFAULT 0 NOT NULL, NAME VARCHAR(255), ROBOT_ID BIGINT NOT NULL, PRIMARY KEY (ID)); CREATE TABLE MC_RESOURCE_FILE (ID BIGINT NOT NULL, BYTESIZE INTEGER NOT NULL, BYTES BLOB(2147483647) NOT NULL, CREATEDBY VARCHAR(255), LASTMODIFIED TIMESTAMP NOT NULL, MODIFIEDBY VARCHAR(255), NAME VARCHAR(255) NOT NULL, SHA_HASH VARCHAR(255) NOT NULL, FOLDER_EX BIGINT NOT NULL, P_ID BIGINT NOT NULL, PRIMARY KEY (ID)); CREATE TABLE MC_PROJECT (ID BIGINT NOT NULL, ACAO VARCHAR(255), AUTHENTICATE SMALLINT DEFAULT 0, DESCRIPTION VARCHAR(255), NAME VARCHAR(255) NOT NULL UNIQUE, CLUSTER_ID BIGINT, PRIMARY KEY (ID)); CREATE TABLE MC_ROBOSERVER_CLUSTER (ID BIGINT NOT NULL, KCU INTEGER, NAME VARCHAR(255) NOT NULL UNIQUE, type SMALLINT DEFAULT 0 NOT NULL, USESSL SMALLINT DEFAULT 0 NOT NULL, SETTINGS_ID BIGINT, PRIMARY KEY (ID));	MC_CONTAINER_BLOCK DROP CONSTRAINT MCCONTAINERBLOCKID; ALTER TABLE MC_SEQUENTIAL_BLOCK DROP CONSTRAINT MCSEQUENTIALBLOCKD; ALTER TABLE MC_CONCURRENT_BLOCK DROP CONSTRAINT MCCONCURRENTBLOCKD; ALTER TABLE MC_SCRIPT_BLOCK DROP CONSTRAINT MC_SCRIPT_BLOCK_ID; ALTER TABLE MC_FOLDER_ITEM DROP CONSTRAINT MCFOLDERITEMFOLDER; ALTER TABLE MC_ROBOT DROP CONSTRAINT FK_MC_ROBOT_ID; ALTER TABLE MC_ROBOT DROP CONSTRAINT MC_ROBOT_FOLDER_EX; ALTER TABLE MC_ROBOT DROP CONSTRAINT FK_MC_ROBOT_P_ID; ALTER TABLE MC_ROBOT DROP CONSTRAINT UNQ_MC_ROBOT_0; ALTER TABLE MC_TYPE DROP CONSTRAINT FK_MC_TYPE_P_ID; ALTER TABLE MC_TYPE DROP CONSTRAINT FK_MC_TYPE_ID; ALTER TABLE MC_TYPE DROP CONSTRAINT UNQ_MC_TYPE_0; ALTER TABLE MC_SNIPPET DROP CONSTRAINT FK_MC_SNIPPET_P_ID; ALTER TABLE MC_SNIPPET DROP CONSTRAINT FK_MC_SNIPPET_ID; ALTER TABLE MC_SNIPPET DROP CONSTRAINT

Database	Create Tables	Drop Tables
	CREATE TABLE MC_GROUP_TO_ROLE_MAP (ID BIGINT NOT NULL, GROUP_NAME VARCHAR(200) NOT NULL, ROLE_NAME VARCHAR(40) NOT NULL, P_ID BIGINT NOT NULL, PRIMARY KEY (ID)); CREATE TABLE MC_SNIPPET_PARAMETER (ID BIGINT NOT NULL, TYPENAME VARCHAR(255) NOT NULL, VARIABLENAME VARCHAR(255) NOT NULL, SNIPPET_ID BIGINT NOT NULL, PRIMARY KEY (ID)); CREATE TABLE MC_SNIPPET_REF_SNIPPET (ID BIGINT NOT NULL, NAME VARCHAR(255) NOT NULL, SNIPPET_ID BIGINT NOT NULL, PRIMARY KEY (ID)); CREATE TABLE MC_SNIPPET_REF_TYPE (ID BIGINT NOT NULL, ISINPUT SMALLINT DEFAULT 0 NOT NULL, ISRETURNED SMALLINT DEFAULT 0 NOT NULL, ISSTORED SMALLINT DEFAULT 0 NOT NULL, NAME VARCHAR(255), SNIPPET_ID BIGINT NOT NULL, PRIMARY KEY (ID)); CREATE TABLE MC_SNIPPET_TYPED_VAR (ID BIGINT NOT NULL, TYPENAME VARCHAR(255) NOT NULL, VARIABLENAME VARCHAR(255) NOT NULL, SNIPPET_ID BIGINT NOT NULL, PRIMARY KEY (ID)); CREATE TABLE MC_SNIPPET_REF_VAR (ID BIGINT NOT NULL, ISRETURNED SMALLINT DEFAULT 0 NOT NULL, ISSTORED SMALLINT DEFAULT 0 NOT NULL, NAME VARCHAR(255), SNIPPET_ID BIGINT NOT NULL, PRIMARY KEY (ID)); CREATE TABLE MC_OAUTH_APPLICATION (ID BIGINT NOT NULL, CALLBACKURL VARCHAR(255), CONSUMERKEY VARCHAR(255), CONSUMERSECRET CLOB(2147483647), NAME VARCHAR(255) NOT NULL, SCOPE VARCHAR(255), SERVICEPROVIDER VARCHAR(255), P_ID BIGINT NOT NULL, PRIMARY KEY (ID)); CREATE TABLE MC_OAUTH_USER (ID BIGINT NOT NULL, ACCESSTOKEN VARCHAR(255), ACCESSTOKENSECRET CLOB(2147483647), NAME VARCHAR(255) NOT NULL, REFRESHTOKEN VARCHAR(255), APPLICATION_ID BIGINT NOT NULL, USER_ID VARCHAR(255), PRIMARY KEY (ID)); CREATE TABLE MC_DSLICENSE (TOKEN VARCHAR(255) NOT NULL, EXPIRATION BIGINT, IP VARCHAR(255), USERNAME VARCHAR(255), PRIMARY KEY (TOKEN)); CREATE TABLE MC_USER (USER_NAME VARCHAR(255) NOT NULL, ADMIN_USER SMALLINT DEFAULT 0, CREATION_DATE TIMESTAMP, EMAIL_ADDR VARCHAR(255), FULL_NAME VARCHAR(255), LAST_IP_ADDR VARCHAR(255), LOGIN_TIME TIMESTAMP, LOGIN_COUNT INTEGER NOT NULL, PASS_WORD VARCHAR(255), PRIMARY KEY (USER_NAME)); CREATE TABLE MC_GROUP (ID BIGINT NOT NULL, DESCRIPTION VARCHAR(255), GROUP_NAME VARCHAR(255) NOT NULL, PRIMARY KEY (ID)); CREATE TABLE MC_USER_GROUP_REL (ID BIGINT NOT NULL, GROUP_ID BIGINT NOT NULL, USER_ID VARCHAR(255) NOT NULL, PRIMARY KEY (ID)); CREATE TABLE MC_USER_PROP (ID BIGINT NOT NULL, IS_NULL SMALLINT DEFAULT 0, PROP_NAME VARCHAR(255) NOT NULL, PROP_VALUE CLOB(2147483647), USER_ID VARCHAR(255)	UNQ_MC_SNIPPET_0; ALTER TABLE MC_ROBOSERVER DROP CONSTRAINT MCRBSERVERCLSTERID; ALTER TABLE MC_ROBOT_PARAMETER DROP CONSTRAINT MCRBTPARAMETERRBTD; ALTER TABLE MC_ROBOT_REF_TYPE DROP CONSTRAINT MCRBOTREFTYPERBTID; ALTER TABLE MC_ROBOT_REF_SNIPPET DROP CONSTRAINT MCRBTRFSNIPPETRBTD; ALTER TABLE MC_ROBOT_TYPED_VARIABLE DROP CONSTRAINT MCRBTTYPDVRBLERBTD; ALTER TABLE MC_ROBOT_REF_VAR DROP CONSTRAINT MCRBOTREFVARRBOTID; ALTER TABLE MC_RESOURCE_FILE DROP CONSTRAINT MC_RESOURCE_FILEID; ALTER TABLE MC_RESOURCE_FILE DROP CONSTRAINT MCRESOURCEFILEP_ID; ALTER TABLE MC_RESOURCE_FILE DROP CONSTRAINT MCRSURCEFILEFLDREX; ALTER TABLE MC_RESOURCE_FILE DROP CONSTRAINT MC_RESOURCE_FILE0; ALTER TABLE MC_PROJECT DROP CONSTRAINT MCPROJECTCLUSTERID; ALTER TABLE MC_ROBOSERVER_CLUSTER DROP CONSTRAINT MCRBSRVRCLSSTTNGSD; ALTER TABLE MC_GROUP_TO_ROLE_MAP DROP CONSTRAINT MCGROUPTOROLEMAPPD; ALTER TABLE MC_GROUP_TO_ROLE_MAP DROP CONSTRAINT MCGROUPTOROLE_MAP0; ALTER TABLE

Database	Create Tables	Drop Tables
	NOT NULL, PRIMARY KEY (ID)); CREATE TABLE MC_TYPE_ATTR (ID BIGINT NOT NULL, ATTR_TYPE VARCHAR(255), COMMENTS CLOB(2147483647), DEFAULT_VALUE CLOB(2147483647), NAME VARCHAR(255), REQUIRED_ATTR SMALLINT DEFAULT 0, TYPE_ORDER INTEGER, TYPE_ID BIGINT NOT NULL, PRIMARY KEY (ID)); CREATE TABLE MC_RS_CLUSTER_SETTINGS (ID BIGINT NOT NULL, CLUSTER_SETTINGS_ID VARCHAR(255) NOT NULL, EMAIL_LOG_FROM_ADDRESS VARCHAR(255), LOG_INPUT_TO_DB SMALLINT DEFAULT 0, LOG_INPUT_TO_LOG4J SMALLINT DEFAULT 0, LOG_TO_DB SMALLINT DEFAULT 0 NOT NULL, LOG_TO_EMAIL SMALLINT DEFAULT 0 NOT NULL, LOG_TO_LOG4J SMALLINT DEFAULT 0 NOT NULL, MAX_CONCURRENT_ROBOTS INTEGER, MAX_QUEUED_ROBOTS INTEGER, PROF_ENABLED SMALLINT DEFAULT 0, PROF_FILE_OUTPUT_APPEND SMALLINT DEFAULT 0, PROF_FILE_OUTPUT_FILE VARCHAR(255), PROF_LOG_PAGE_URL SMALLINT DEFAULT 0, PROF_OUTPUT_TARGET VARCHAR(20), PROF_TYPE VARCHAR(20), PROF_THRESHOLD INTEGER, EMAIL_LOG_TO_ADDRESS VARCHAR(255), PRIMARY KEY (ID)); CREATE TABLE MC_DATABASE (ID BIGINT NOT NULL, HOST VARCHAR(255) NOT NULL, MAX_ACTIVE_CONNECTIONS INTEGER, MAX_IDLE_CONNECTIONS INTEGER, NAME VARCHAR(255) NOT NULL, PASSWORD VARCHAR(255), SCHEMA_NAME VARCHAR(255) NOT NULL, TYPE VARCHAR(255) NOT NULL, USERNAME VARCHAR(255), CLUSTER_SETTINGS_ID BIGINT NOT NULL, PRIMARY KEY (ID)); CREATE TABLE MC_PROXY_SERVER (ID BIGINT NOT NULL, EXCLUDED_HOST_NAMES VARCHAR(1000), HOST_NAME VARCHAR(255) NOT NULL, PASSWORD VARCHAR(255), PORT_NUMBER INTEGER, USERNAME VARCHAR(255), CLUSTER_SETTINGS_ID BIGINT NOT NULL, PRIMARY KEY (ID)); CREATE TABLE MC_DATABASE_TYPE (ID BIGINT NOT NULL, DRIVER_CLASS VARCHAR(255) NOT NULL, NAME VARCHAR(255) NOT NULL, URL_TEMPLATE VARCHAR(255) NOT NULL, VALIDATION_QUERY VARCHAR(255) NOT NULL, PRIMARY KEY (ID)); CREATE TABLE MC_JAR_FILE (ID BIGINT NOT NULL, BYTES BLOB(2147483647), CREATED TIMESTAMP, NAME VARCHAR(255) NOT NULL, FILE_SIZE INTEGER, PRIMARY KEY (ID)); CREATE TABLE APP_KAPPLET (ID VARCHAR(40) NOT NULL, DTYPE VARCHAR(31), CREATED_DATE TIMESTAMP NOT NULL, DESCRIPTION CLOB(2147483647), NAME VARCHAR(255) NOT NULL, OPTIONS CLOB(2147483647), P_ID BIGINT NOT NULL, ICON_ID VARCHAR(255), PRIMARY KEY (ID)); CREATE TABLE APP_MASTER (ID VARCHAR(40) NOT NULL, CREATEDBY VARCHAR(255), ENABLED SMALLINT DEFAULT 0 NOT NULL, FLOW CLOB(2147483647), INITIAL_LABEL INTEGER, KAPPLET_TYPE	MC_SNIPPET_PARAMETER DROP CONSTRAINT MCSNPPTPRMTRSNPPTD; ALTER TABLE MC_SNIPPET_REF_SNIPPET DROP CONSTRAINT MCSNPPTRFSNPSNPPTD; ALTER TABLE MC_SNIPPET_REF_TYPE DROP CONSTRAINT MCSNPPTRFTYPSNPPTD; ALTER TABLE MC_SNIPPET_TYPED_VAR DROP CONSTRAINT MCSNPPTTYPDVSNPPTD; ALTER TABLE MC_SNIPPET_REF_VAR DROP CONSTRAINT MCSNPPTRFVARSNPPTD; ALTER TABLE MC_OAUTH_APPLICATION DROP CONSTRAINT MCUTHAPPLICATIONPD; ALTER TABLE MC_OAUTH_APPLICATION DROP CONSTRAINT MCAUTHAPPLICATION0; ALTER TABLE MC_OAUTH_USER DROP CONSTRAINT MCOAUTHUSERUSER_ID; ALTER TABLE MC_OAUTH_USER DROP CONSTRAINT MCTHUSERPPLCTIONID; ALTER TABLE MC_OAUTH_USER DROP CONSTRAINT MC_OAUTH_USER0; ALTER TABLE MC_GROUP DROP CONSTRAINT UNQ_MC_GROUP_0; ALTER TABLE MC_USER_GROUP_REL DROP CONSTRAINT MCSERGROUPRELSERID; ALTER TABLE MC_USER_GROUP_REL DROP CONSTRAINT MCSERGROUPRELGRPID; ALTER TABLE MC_USER_GROUP_REL DROP CONSTRAINT MC_USER_GROUP_REL0; ALTER TABLE MC_USER_PROP DROP CONSTRAINT

Database	Create Tables	Drop Tables
	VARCHAR(255) NOT NULL, LAST_INSTALLED TIMESTAMP, LAST_MODIFIED TIMESTAMP NOT NULL, LAST_RUN TIMESTAMP, MODIFIEDBY VARCHAR(255), PRIMARY KEY (ID)); CREATE TABLE APP_INSTALLED (ID VARCHAR(40) NOT NULL, FAVORITE SMALLINT DEFAULT 0, LAST_RUN TIMESTAMP, UPDATED_AT TIMESTAMP, PARENT VARCHAR(40) NOT NULL, USER_ID VARCHAR(255) NOT NULL, PRIMARY KEY (ID)); CREATE TABLE APP_ICON (ID VARCHAR(255) NOT NULL, ICON BLOB(2147483647), filename VARCHAR(255), mimeType VARCHAR(255), PRIMARY KEY (ID)); CREATE TABLE APP_UPLOAD (ID VARCHAR(255) NOT NULL, UPLOAD BLOB(2147483647) NOT NULL, filename VARCHAR(255) NOT NULL, mimeType VARCHAR(255) NOT NULL, timestamp TIMESTAMP NOT NULL, username VARCHAR(255) NOT NULL, PRIMARY KEY (ID)); CREATE TABLE MC_KAPPLET_PROP (ID BIGINT NOT NULL, IS_NULL SMALLINT DEFAULT 0, PROP_NAME VARCHAR(255) NOT NULL, PROP_VALUE CLOB(2147483647), KAPPLET_ID VARCHAR(40) NOT NULL, PRIMARY KEY (ID)); CREATE TABLE APP_SCHEDULE_ERROR (ID BIGINT NOT NULL, ERROR CLOB(2147483647), SCHEDULE_ID BIGINT NOT NULL, PRIMARY KEY (ID)); CREATE TABLE APP_SCHEDULE (ID BIGINT NOT NULL, DIRTY SMALLINT DEFAULT 0, LASTRUN TIMESTAMP, NEXTRUN TIMESTAMP, PARAMETERS CLOB(2147483647), APP_ID VARCHAR(40) NOT NULL, PRIMARY KEY (ID)); CREATE TABLE APP_SCHEDULE_VALUE (ID VARCHAR(255) NOT NULL, VALUE BLOB(2147483647) NOT NULL, SCHEDULE_ID BIGINT NOT NULL, PRIMARY KEY (ID)); CREATE TABLE MC_GLOBAL_KAPPLET_PROP (PROP_NAME VARCHAR(255) NOT NULL, IS_NULL SMALLINT DEFAULT 0, PROP_VALUE CLOB(2147483647), PRIMARY KEY (PROP_NAME)); CREATE TABLE APP_ACTION_EXECUTION (ID BIGINT NOT NULL, END_TIME TIMESTAMP, LABEL INTEGER NOT NULL, SEQ_ID INTEGER, START_TIME TIMESTAMP, STATUS VARCHAR(255), TASK_IDS VARCHAR(255), VERSIONLOCK BIGINT, KAPPLET_RUN_ID BIGINT NOT NULL, PRIMARY KEY (ID)); CREATE TABLE APP_DATA_ROW (ID BIGINT NOT NULL, TABLE_ID BIGINT NOT NULL, PRIMARY KEY (ID)); CREATE TABLE APP_DATA_TABLE (ID BIGINT NOT NULL, TABLE_NAME VARCHAR(255) NOT NULL, FLOW_EXECUTION_ID BIGINT NOT NULL, PRIMARY KEY (ID)); CREATE TABLE APP_DATA_STR (ID BIGINT NOT NULL, NAME VARCHAR(255), VALUE VARCHAR(2000) NOT NULL, RESULT_ID BIGINT NOT NULL, PRIMARY KEY (ID)); CREATE TABLE APP_DATA_LOB (ID BIGINT NOT NULL, LARGE SMALLINT DEFAULT 0, NAME VARCHAR(255), VALUE CLOB(2147483647), RESULT_ID BIGINT NOT NULL, PRIMARY KEY (ID)); CREATE TABLE APP_FLOW_ERROR (ID BIGINT NOT NULL, ERROR CLOB(2147483647),	MCUSER_PROPUSER_ID; ALTER TABLE MC_USER_PROP DROP CONSTRAINT UNQ_MC_USER_PROP_0; ALTER TABLE MC_TYPE_ATTR DROP CONSTRAINT MCTYPE_ATTRTYPE_ID; ALTER TABLE MC_DATABASE DROP CONSTRAINT MCDTBSCLSTRSTTNGSD; ALTER TABLE MC_PROXY_SERVER DROP CONSTRAINT MCPRXYCLSTRSTTNGSD; ALTER TABLE APP_KAPPLET DROP CONSTRAINT APP_KAPPLET_P_ID; ALTER TABLE APP_KAPPLET DROP CONSTRAINT APP_KAPPLETICON_ID; ALTER TABLE APP_MASTER DROP CONSTRAINT FK_APP_MASTER_ID; ALTER TABLE APP_INSTALLED DROP CONSTRAINT APPINSTALLEDPARENT; ALTER TABLE APP_INSTALLED DROP CONSTRAINT APP_INSTALLED_ID; ALTER TABLE APP_INSTALLED DROP CONSTRAINT APPINSTALLEDUSERID; ALTER TABLE MC_KAPPLET_PROP DROP CONSTRAINT MCKPPLTPROPKPPLTID; ALTER TABLE MC_KAPPLET_PROP DROP CONSTRAINT MC_KAPPLET_PROP0; ALTER TABLE APP_SCHEDULE_ERROR DROP CONSTRAINT PPSCHDLERRORSCHDLD; ALTER TABLE APP_SCHEDULE DROP CONSTRAINT APP_SCHEDULEAPP_ID; ALTER TABLE

Database	Create Tables	Drop Tables
	EXECUTION_ID BIGINT NOT NULL, PRIMARY KEY (ID)); CREATE TABLE APP_RUN (ID BIGINT NOT NULL, ABORTED SMALLINT DEFAULT 0, FLOW CLOB(2147483647), INITIAL_LABEL INTEGER NOT NULL, LASTEXECUTIONPATHCHANGE BIGINT, OPTIONS CLOB(2147483647), PARAMETERS CLOB(2147483647), SIGNATURES CLOB(2147483647), VERSIONLOCK BIGINT, KAPPLET_ID VARCHAR(40) NOT NULL, PRIMARY KEY (ID)); CREATE TABLE APP_RUN_VALUE (ID VARCHAR(255) NOT NULL, VALUE BLOB(2147483647) NOT NULL, RUN_ID BIGINT NOT NULL, PRIMARY KEY (ID)); CREATE TABLE MC_FOLDER (ID BIGINT NOT NULL, DTYPE VARCHAR(31), PATH VARCHAR(255), PRIMARY KEY (ID)); CREATE TABLE MC_ROOT_FOLDER (ID BIGINT NOT NULL, P_ID BIGINT NOT NULL, PRIMARY KEY (ID)); CREATE TABLE MC_SUBFOLDER (ID BIGINT NOT NULL, NAME VARCHAR(255) NOT NULL, PARENT_ID BIGINT NOT NULL, PRIMARY KEY (ID)); CREATE TABLE APP_RUN_APP_SCHEDULE (schedules_ID BIGINT NOT NULL, kappletRuns_ID BIGINT NOT NULL, PRIMARY KEY (schedules_ID, kappletRuns_ID)); ALTER TABLE MC_SCHEDULE ADD CONSTRAINT UNQ_MC_SCHEDULE_0 UNIQUE (P_ID, NAME); ALTER TABLE MC_BLOCK ADD CONSTRAINT UNQ_MC_BLOCK_0 UNIQUE (P_ID, NAME); ALTER TABLE MC_BLOCK_REFERENCE ADD CONSTRAINT MCBLOCK_REFERENCE0 UNIQUE (PARENT, NAME); ALTER TABLE MC_ROBOT ADD CONSTRAINT UNQ_MC_ROBOT_0 UNIQUE (P_ID, NAME, FOLDER_EX); ALTER TABLE MC_TYPE ADD CONSTRAINT UNQ_MC_TYPE_0 UNIQUE (P_ID, NAME); ALTER TABLE MC_SNIPPET ADD CONSTRAINT UNQ_MC_SNIPPET_0 UNIQUE (P_ID, NAME); ALTER TABLE MC_RESOURCE_FILE ADD CONSTRAINT MC_RESOURCE_FILE0 UNIQUE (P_ID, NAME, FOLDER_EX); ALTER TABLE MC_GROUP_TO_ROLE_MAP ADD CONSTRAINT MCGROUPTOROLE_MAP0 UNIQUE (P_ID, ROLE_NAME, GROUP_NAME); ALTER TABLE MC_OAUTH_APPLICATION ADD CONSTRAINT MCAUTHAPPLICATION0 UNIQUE (P_ID, NAME); ALTER TABLE MC_OAUTH_USER ADD CONSTRAINT MC_OAUTH_USER0 UNIQUE (APPLICATION_ID, NAME); ALTER TABLE MC_GROUP ADD CONSTRAINT UNQ_MC_GROUP_0 UNIQUE (GROUP_NAME); ALTER TABLE MC_USER_GROUP_REL ADD CONSTRAINT MC_USER_GROUP_REL0 UNIQUE (USER_ID, GROUP_ID); ALTER TABLE MC_USER_PROP ADD CONSTRAINT UNQ_MC_USER_PROP_0 UNIQUE (ID, PROP_NAME); ALTER TABLE MC_KAPPLET_PROP ADD CONSTRAINT MC_KAPPLET_PROP0 UNIQUE (ID, PROP_NAME); ALTER TABLE APP_ACTION_EXECUTION ADD CONSTRAINT PPACTIONEXECUTION0 UNIQUE (KAPPLET_RUN_ID, LABEL); ALTER TABLE APP_DATA_TABLE ADD CONSTRAINT APP_DATA_TABLE0 UNIQUE (FLOW_EXECUTION_ID,	APP_SCHEDULE_VALUE DROP CONSTRAINT PPSCHDLVALUESCHDLD; ALTER TABLE APP_ACTION_EXECUTION DROP CONSTRAINT PPCTNXCTONKPPLTRND; ALTER TABLE APP_ACTION_EXECUTION DROP CONSTRAINT PPACTIONEXECUTION0; ALTER TABLE APP_DATA_ROW DROP CONSTRAINT APPDATAROWTABLE_ID; ALTER TABLE APP_DATA_TABLE DROP CONSTRAINT PPDTTBLEFLWXCTONID; ALTER TABLE APP_DATA_TABLE DROP CONSTRAINT APP_DATA_TABLE0; ALTER TABLE APP_DATA_STR DROP CONSTRAINT APPDATASTRRESULTID; ALTER TABLE APP_DATA_LOB DROP CONSTRAINT APPDATALOBRESULTID; ALTER TABLE APP_FLOW_ERROR DROP CONSTRAINT PPFLWERRORXCTIONID; ALTER TABLE APP_RUN DROP CONSTRAINT APP_RUN_KAPPLET_ID; ALTER TABLE APP_RUN_VALUE DROP CONSTRAINT APPRUN_VALUERUN_ID; ALTER TABLE MC_ROOT_FOLDER DROP CONSTRAINT MC_ROOT_FOLDERP_ID; ALTER TABLE MC_ROOT_FOLDER DROP CONSTRAINT MC_ROOT_FOLDER_ID; ALTER TABLE MC_SUBFOLDER DROP CONSTRAINT MCSUBFOLDERPRENTID; ALTER TABLE MC_SUBFOLDER DROP CONSTRAINT

Database	Create Tables	Drop Tables
	TABLE_NAME); ALTER TABLE MC_SUBFOLDER ADD CONSTRAINT UNQ_MC_SUBFOLDER_0 UNIQUE (NAME, PARENT_ID); ALTER TABLE MC_SCHEDULE ADD CONSTRAINT MCSCHDULEBLCKJOBID FOREIGN KEY (BLOCKJOB_ID) REFERENCES MC_BLOCK_JOB (ID); ALTER TABLE MC_SCHEDULE ADD CONSTRAINT MC_SCHEDULE_P_ID FOREIGN KEY (P_ID) REFERENCES MC_PROJECT (ID); ALTER TABLE MC_SCHEDULE ADD CONSTRAINT MCSCHEDULECLSTERID FOREIGN KEY (CLUSTER_ID) REFERENCES MC_ROBOSERVER_CLUSTER (ID); ALTER TABLE MC_BLOCK ADD CONSTRAINT FK_MC_BLOCK_P_ID FOREIGN KEY (P_ID) REFERENCES MC_PROJECT (ID); ALTER TABLE MC_BLOCK_REFERENCE ADD CONSTRAINT MCBLCKRFERENCECHLD FOREIGN KEY (CHILD) REFERENCES MC_BLOCK (ID); ALTER TABLE MC_BLOCK_REFERENCE ADD CONSTRAINT MCBLCKRFERENCEPRNT FOREIGN KEY (PARENT) REFERENCES MC_BLOCK (ID); ALTER TABLE MC_BLOCK_JOB ADD CONSTRAINT MCBLOCKJOBBLOCK_ID FOREIGN KEY (BLOCK_ID) REFERENCES MC_BLOCK (ID); ALTER TABLE MC_ROBOT_BLOCK ADD CONSTRAINT MC_ROBOT_BLOCK_ID FOREIGN KEY (ID) REFERENCES MC_BLOCK (ID); ALTER TABLE MC_ROBOT_BLOCK ADD CONSTRAINT MCROBOTBLOCKRBOTID FOREIGN KEY (ROBOT_ID) REFERENCES MC_FOLDER_ITEM (ID); ALTER TABLE MC_MULTIPLE_ROBOT_BLOCK ADD CONSTRAINT MCMLTPLROBOTBLOCKD FOREIGN KEY (ID) REFERENCES MC_BLOCK (ID); ALTER TABLE MC_CONTAINER_BLOCK ADD CONSTRAINT MCCONTAINERBLOCKID FOREIGN KEY (ID) REFERENCES MC_BLOCK (ID); ALTER TABLE MC_SEQUENTIAL_BLOCK ADD CONSTRAINT MCSEQUENTIALBLOCKD FOREIGN KEY (ID) REFERENCES MC_BLOCK (ID); ALTER TABLE MC_CONCURRENT_BLOCK ADD CONSTRAINT MCCONCURRENTBLOCKD FOREIGN KEY (ID) REFERENCES MC_BLOCK (ID); ALTER TABLE MC_SCRIPT_BLOCK ADD CONSTRAINT MC_SCRIPT_BLOCK_ID FOREIGN KEY (ID) REFERENCES MC_BLOCK (ID); ALTER TABLE MC_FOLDER_ITEM ADD CONSTRAINT MCFOLDERITEMFOLDER FOREIGN KEY (FOLDER) REFERENCES MC_FOLDER (ID); ALTER TABLE MC_ROBOT ADD CONSTRAINT FK_MC_ROBOT_ID FOREIGN KEY (ID) REFERENCES MC_FOLDER_ITEM (ID); ALTER TABLE MC_ROBOT ADD CONSTRAINT MC_ROBOT_FOLDER_EX FOREIGN KEY (FOLDER_EX) REFERENCES MC_FOLDER (ID); ALTER TABLE MC_ROBOT ADD CONSTRAINT FK_MC_ROBOT_P_ID FOREIGN KEY (P_ID) REFERENCES MC_PROJECT (ID); ALTER TABLE MC_TYPE ADD CONSTRAINT FK_MC_TYPE_P_ID FOREIGN KEY (P_ID) REFERENCES MC_PROJECT (ID); ALTER TABLE MC_TYPE ADD CONSTRAINT FK_MC_TYPE_ID	FK_MC_SUBFOLDER_ID; ALTER TABLE MC_SUBFOLDER DROP CONSTRAINT UNQ_MC_SUBFOLDER_0; ALTER TABLE APP_RUN_APP_SCHEDULE DROP CONSTRAINT PPRNPPSCHDLschdlsD; ALTER TABLE APP_RUN_APP_SCHEDULE DROP CONSTRAINT PPRNPPSCHkppltRnsD; DROP TABLE MC_SCHEDULE; DROP TABLE MC_BLOCK; DROP TABLE MC_BLOCK_REFERENCE; DROP TABLE MC_BLOCK_JOB; DROP TABLE MC_ROBOT_BLOCK; DROP TABLE MC_MULTIPLE_ROBOT_BLOCK; DROP TABLE MC_CONTAINER_BLOCK; DROP TABLE MC_SEQUENTIAL_BLOCK; DROP TABLE MC_CONCURRENT_BLOCK; DROP TABLE MC_SCRIPT_BLOCK; DROP TABLE MC_FOLDER_ITEM; DROP TABLE MC_ROBOT; DROP TABLE MC_TYPE; DROP TABLE MC_SNIPPET; DROP TABLE MC_ROBOSERVER; DROP TABLE MC_SETTINGS; DROP TABLE MC_ROBOT_PARAMETER; DROP TABLE MC_ROBOT_REF_TYPE; DROP TABLE MC_ROBOT_REF_SNIPPET; DROP TABLE MC_ROBOT_TYPED_VARIABLE; DROP TABLE MC_ROBOT_REF_VAR; DROP TABLE MC_RESOURCE_FILE; DROP TABLE MC_PROJECT; DROP TABLE

Database	Create Tables	Drop Tables
	FOREIGN KEY (ID) REFERENCES MC_FOLDER_ITEM (ID); ALTER TABLE MC_SNIPPET ADD CONSTRAINT FK_MC_SNIPPET_P_ID FOREIGN KEY (P_ID) REFERENCES MC_PROJECT (ID); ALTER TABLE MC_SNIPPET ADD CONSTRAINT FK_MC_SNIPPET_ID FOREIGN KEY (ID) REFERENCES MC_FOLDER_ITEM (ID); ALTER TABLE MC_ROBOSERVER ADD CONSTRAINT MCRBSERVERCLSTERID FOREIGN KEY (CLUSTER_ID) REFERENCES MC_ROBOSERVER_CLUSTER (ID); ALTER TABLE MC_ROBOT_PARAMETER ADD CONSTRAINT MCRBTPARAMETERRBTD FOREIGN KEY (ROBOT_ID) REFERENCES MC_FOLDER_ITEM (ID); ALTER TABLE MC_ROBOT_REF_TYPE ADD CONSTRAINT MCRBOTREFTYPERBTID FOREIGN KEY (ROBOT_ID) REFERENCES MC_FOLDER_ITEM (ID); ALTER TABLE MC_ROBOT_REF_SNIPPET ADD CONSTRAINT MCRBTRFSNIPPETRBTD FOREIGN KEY (ROBOT_ID) REFERENCES MC_FOLDER_ITEM (ID); ALTER TABLE MC_ROBOT_TYPED_VARIABLE ADD CONSTRAINT MCRBTTYPDVRBLERBTD FOREIGN KEY (ROBOT_ID) REFERENCES MC_FOLDER_ITEM (ID); ALTER TABLE MC_ROBOT_REF_VAR ADD CONSTRAINT MCRBOTREFVARRBOTID FOREIGN KEY (ROBOT_ID) REFERENCES MC_FOLDER_ITEM (ID); ALTER TABLE MC_RESOURCE_FILE ADD CONSTRAINT MC_RESOURCE_FILEID FOREIGN KEY (ID) REFERENCES MC_FOLDER_ITEM (ID); ALTER TABLE MC_RESOURCE_FILE ADD CONSTRAINT MCRESOURCEFILEP_ID FOREIGN KEY (P_ID) REFERENCES MC_PROJECT (ID); ALTER TABLE MC_RESOURCE_FILE ADD CONSTRAINT MCRSURCEFILEFLDREX FOREIGN KEY (FOLDER_EX) REFERENCES MC_FOLDER (ID); ALTER TABLE MC_PROJECT ADD CONSTRAINT MCPROJECTCLUSTERID FOREIGN KEY (CLUSTER_ID) REFERENCES MC_ROBOSERVER_CLUSTER (ID); ALTER TABLE MC_ROBOSERVER_CLUSTER ADD CONSTRAINT MCRBSRVRCLSSTTNGSD FOREIGN KEY (SETTINGS_ID) REFERENCES MC_RS_CLUSTER_SETTINGS (ID); ALTER TABLE MC_GROUP_TO_ROLE_MAP ADD CONSTRAINT MCGROUPTOROLEMAPPD FOREIGN KEY (P_ID) REFERENCES MC_PROJECT (ID); ALTER TABLE MC_SNIPPET_PARAMETER ADD CONSTRAINT MCSNPPTPRMTRSNPPTD FOREIGN KEY (SNIPPET_ID) REFERENCES MC_FOLDER_ITEM (ID); ALTER TABLE MC_SNIPPET_REF_SNIPPET ADD CONSTRAINT MCSNPPTRFSNPSNPPTD FOREIGN KEY (SNIPPET_ID) REFERENCES MC_FOLDER_ITEM (ID); ALTER TABLE MC_SNIPPET_REF_TYPE ADD CONSTRAINT MCSNPPTRFTYPSNPPTD FOREIGN KEY (SNIPPET_ID) REFERENCES MC_FOLDER_ITEM (ID); ALTER TABLE MC_SNIPPET_TYPED_VAR ADD CONSTRAINT MCSNPPTTYPDVSNPPTD FOREIGN KEY (SNIPPET_ID) REFERENCES MC_FOLDER_ITEM (ID); ALTER TABLE MC_SNIPPET_REF_VAR ADD CONSTRAINT MCSNPPTRFVARSNPPTD FOREIGN KEY (SNIPPET_ID) REFERENCES MC_FOLDER_ITEM	MC_ROBOSERVER_CLUSTER; DROP TABLE MC_GROUP_TO_ROLE_MAP; DROP TABLE MC_SNIPPET_PARAMETER; DROP TABLE MC_SNIPPET_REF_SNIPPET; DROP TABLE MC_SNIPPET_REF_TYPE; DROP TABLE MC_SNIPPET_TYPED_VAR; DROP TABLE MC_SNIPPET_REF_VAR; DROP TABLE MC_OAUTH_APPLICATION; DROP TABLE MC_OAUTH_USER; DROP TABLE MC_DSLICENSE; DROP TABLE MC_USER; DROP TABLE MC_GROUP; DROP TABLE MC_USER_GROUP_REL; DROP TABLE MC_USER_PROP; DROP TABLE MC_TYPE_ATTR; DROP TABLE MC_RS_CLUSTER_SETTINGS; DROP TABLE MC_DATABASE; DROP TABLE MC_PROXY_SERVER; DROP TABLE MC_DATABASE_TYPE; DROP TABLE MC_JAR_FILE; DROP TABLE APP_KAPPLET; DROP TABLE APP_MASTER; DROP TABLE APP_INSTALLED; DROP TABLE APP_ICON; DROP TABLE APP_UPLOAD; DROP TABLE MC_KAPPLET_PROP; DROP TABLE APP_SCHEDULE_ERROR; DROP TABLE APP_SCHEDULE; DROP TABLE APP_SCHEDULE_VALUE; DROP TABLE MC_GLOBAL_KAPPLET_PROP; DROP TABLE APP_ACTION_EXECUTION;

Database	Create Tables	Drop Tables
	(ID); ALTER TABLE MC_OAUTH_APPLICATION ADD CONSTRAINT MCUTHAPPLICATIONPD FOREIGN KEY (P_ID) REFERENCES MC_PROJECT (ID); ALTER TABLE MC_OAUTH_USER ADD CONSTRAINT MCOAUTHUSERUSER_ID FOREIGN KEY (USER_ID) REFERENCES MC_USER (USER_NAME); ALTER TABLE MC_OAUTH_USER ADD CONSTRAINT MCTHUSERPPLCTIONID FOREIGN KEY (APPLICATION_ID) REFERENCES MC_OAUTH_APPLICATION (ID); ALTER TABLE MC_USER_GROUP_REL ADD CONSTRAINT MCSERGROUPRELSERID FOREIGN KEY (USER_ID) REFERENCES MC_USER (USER_NAME); ALTER TABLE MC_USER_GROUP_REL ADD CONSTRAINT MCSERGROUPRELGRPID FOREIGN KEY (GROUP_ID) REFERENCES MC_GROUP (ID); ALTER TABLE MC_USER_PROP ADD CONSTRAINT MCUSER_PROPUSER_ID FOREIGN KEY (USER_ID) REFERENCES MC_USER (USER_NAME); ALTER TABLE MC_TYPE_ATTR ADD CONSTRAINT MCTYPE_ATTRTYPE_ID FOREIGN KEY (TYPE_ID) REFERENCES MC_FOLDER_ITEM (ID); ALTER TABLE MC_DATABASE ADD CONSTRAINT MCDTBSCLSTRSTTNGSD FOREIGN KEY (CLUSTER_SETTINGS_ID) REFERENCES MC_RS_CLUSTER_SETTINGS (ID); ALTER TABLE MC_PROXY_SERVER ADD CONSTRAINT MCPRXYCLSTRSTTNGSD FOREIGN KEY (CLUSTER_SETTINGS_ID) REFERENCES MC_RS_CLUSTER_SETTINGS (ID); ALTER TABLE APP_KAPPLET ADD CONSTRAINT APP_KAPPLET_P_ID FOREIGN KEY (P_ID) REFERENCES MC_PROJECT (ID); ALTER TABLE APP_KAPPLET ADD CONSTRAINT APP_KAPPLETICON_ID FOREIGN KEY (ICON_ID) REFERENCES APP_ICON (ID); ALTER TABLE APP_MASTER ADD CONSTRAINT FK_APP_MASTER_ID FOREIGN KEY (ID) REFERENCES APP_KAPPLET (ID); ALTER TABLE APP_INSTALLED ADD CONSTRAINT APPINSTALLEDPARENT FOREIGN KEY (PARENT) REFERENCES APP_KAPPLET (ID); ALTER TABLE APP_INSTALLED ADD CONSTRAINT APP_INSTALLED_ID FOREIGN KEY (ID) REFERENCES APP_KAPPLET (ID); ALTER TABLE APP_INSTALLED ADD CONSTRAINT APPINSTALLEDUSERID FOREIGN KEY (USER_ID) REFERENCES MC_USER (USER_NAME); ALTER TABLE MC_KAPPLET_PROP ADD CONSTRAINT MCKPPLTPROPKPPLTID FOREIGN KEY (KAPPLET_ID) REFERENCES APP_KAPPLET (ID); ALTER TABLE APP_SCHEDULE_ERROR ADD CONSTRAINT PPSCHDLERRORSCHDLD FOREIGN KEY (SCHEDULE_ID) REFERENCES APP_SCHEDULE (ID); ALTER TABLE APP_SCHEDULE ADD CONSTRAINT APP_SCHEDULEAPP_ID FOREIGN KEY (APP_ID) REFERENCES APP_KAPPLET (ID); ALTER TABLE APP_SCHEDULE_VALUE ADD CONSTRAINT PPSCHDLVALUESCHDLD FOREIGN KEY (SCHEDULE_ID) REFERENCES APP_SCHEDULE (ID); ALTER TABLE APP_ACTION_EXECUTION	DROP TABLE APP_DATA_ROW; DROP TABLE APP_DATA_TABLE; DROP TABLE APP_DATA_STR; DROP TABLE APP_DATA_LOB; DROP TABLE APP_FLOW_ERROR; DROP TABLE APP_RUN; DROP TABLE APP_RUN_VALUE; DROP TABLE MC_FOLDER; DROP TABLE MC_ROOT_FOLDER; DROP TABLE MC_SUBFOLDER; DROP TABLE APP_RUN_APP_SCHEDULE; DROP SEQUENCE SEQ_GEN_SEQUENCE RESTRICT; DROP TABLE APP_SEQUENCES;

Database	Create Tables	Drop Tables
	ADD CONSTRAINT PPCTNXCTONKPPLTRND FOREIGN KEY (KAPPLET_RUN_ID) REFERENCES APP_RUN (ID); ALTER TABLE APP_DATA_ROW ADD CONSTRAINT APPDATAROWTABLE_ID FOREIGN KEY (TABLE_ID) REFERENCES APP_DATA_TABLE (ID); ALTER TABLE APP_DATA_TABLE ADD CONSTRAINT PPDTTBLEFLWXCTONID FOREIGN KEY (FLOW_EXECUTION_ID) REFERENCES APP_ACTION_EXECUTION (ID); ALTER TABLE APP_DATA_STR ADD CONSTRAINT APPDATASTRRESULTID FOREIGN KEY (RESULT_ID) REFERENCES APP_DATA_ROW (ID); ALTER TABLE APP_DATA_LOB ADD CONSTRAINT APPDATALOBRESULTID FOREIGN KEY (RESULT_ID) REFERENCES APP_DATA_ROW (ID); ALTER TABLE APP_FLOW_ERROR ADD CONSTRAINT PPFLWERRORXCTIONID FOREIGN KEY (EXECUTION_ID) REFERENCES APP_ACTION_EXECUTION (ID); ALTER TABLE APP_RUN ADD CONSTRAINT APP_RUN_KAPPLET_ID FOREIGN KEY (KAPPLET_ID) REFERENCES APP_KAPPLET (ID); ALTER TABLE APP_RUN_VALUE ADD CONSTRAINT APPRUN_VALUERUN_ID FOREIGN KEY (RUN_ID) REFERENCES APP_RUN (ID); ALTER TABLE MC_ROOT_FOLDER ADD CONSTRAINT MC_ROOT_FOLDERP_ID FOREIGN KEY (P_ID) REFERENCES MC_PROJECT (ID); ALTER TABLE MC_ROOT_FOLDER ADD CONSTRAINT MC_ROOT_FOLDER_ID FOREIGN KEY (ID) REFERENCES MC_FOLDER (ID); ALTER TABLE MC_SUBFOLDER ADD CONSTRAINT MCSUBFOLDERPRENTID FOREIGN KEY (PARENT_ID) REFERENCES MC_FOLDER (ID); ALTER TABLE MC_SUBFOLDER ADD CONSTRAINT FK_MC_SUBFOLDER_ID FOREIGN KEY (ID) REFERENCES MC_FOLDER (ID); ALTER TABLE APP_RUN_APP_SCHEDULE ADD CONSTRAINT PPRNPPSCHDLschdlsD FOREIGN KEY (schedules_ID) REFERENCES APP_SCHEDULE (ID); ALTER TABLE APP_RUN_APP_SCHEDULE ADD CONSTRAINT PPRNPPSCHkppltRnsD FOREIGN KEY (kappletRuns_ID) REFERENCES APP_RUN (ID); CREATE TABLE APP_SEQUENCES (TABLE_NAME VARCHAR(50) NOT NULL, NEXT_ID DECIMAL(15), PRIMARY KEY (TABLE_NAME)); INSERT INTO APP_SEQUENCES(TABLE_NAME, NEXT_ID) values ('DATA_ROW', 0); INSERT INTO APP_SEQUENCES(TABLE_NAME, NEXT_ID) values ('DATA_STR', 0); INSERT INTO APP_SEQUENCES(TABLE_NAME, NEXT_ID) values ('FLOW_ACTION_EXECUTION', 0); INSERT INTO APP_SEQUENCES(TABLE_NAME, NEXT_ID) values ('DATA_TABLE', 0); INSERT INTO APP_SEQUENCES(TABLE_NAME, NEXT_ID) values ('DATA_LOB', 0); CREATE SEQUENCE SEQ_GEN_SEQUENCE INCREMENT BY 50 START WITH 50;	

Database	Create Tables	Drop Tables
MySQL	CREATE TABLE MC_SCHEDULE (ID BIGINT NOT NULL, BAD_INPUT TINYINT(1) default 0, ACTIVE TINYINT(1) default 0, CREATEDBY VARCHAR(255), DIRTY TINYINT(1) default 0, EMAILS VARCHAR(255), MAXOBJECTSEXTRACTED INTEGER, MAXRUNTIME INTEGER, MODIFIEDBY VARCHAR(255), NAME VARCHAR(255) NOT NULL, NEXTRUN DATETIME, PREVIOUSRUN DATETIME, TOTALRUNS INTEGER, USEEMAILNOTIFICATION TINYINT(1) default 0, CLUSTER_ID BIGINT, P_ID BIGINT NOT NULL, BLOCKJOB_ID BIGINT, PRIMARY KEY (ID)); CREATE TABLE MC_BLOCK (ID BIGINT AUTO_INCREMENT NOT NULL, DTYPE VARCHAR(31), NAME VARCHAR(255) NOT NULL, NAMEDBLOCK TINYINT(1) default 0, P_ID BIGINT NOT NULL, PRIMARY KEY (ID)); CREATE TABLE MC_BLOCK_REFERENCE (ID BIGINT AUTO_INCREMENT NOT NULL, ENABLED TINYINT(1) default 0 NOT NULL, ORDER_VAL INTEGER NOT NULL, NAME VARCHAR(255) NOT NULL, CHILD BIGINT, PARENT BIGINT NOT NULL, PRIMARY KEY (ID)); CREATE TABLE MC_BLOCK_JOB (ID BIGINT AUTO_INCREMENT NOT NULL, ACTIVE TINYINT(1) default 0 NOT NULL, BLOCKINPUT LONGTEXT, DISPLAY_NAMES LONGTEXT NOT NULL, BLOCK_ID BIGINT NOT NULL, PRIMARY KEY (ID)); CREATE TABLE MC_ROBOT_BLOCK (ID BIGINT NOT NULL, ROBOT_ID BIGINT, PRIMARY KEY (ID)); CREATE TABLE MC_MULTIPLE_ROBOT_BLOCK (ID BIGINT NOT NULL, DISPLAY_NAME VARCHAR(255), STRATEGY VARCHAR(255), VALUE VARCHAR(255), PRIMARY KEY (ID)); CREATE TABLE MC_CONTAINER_BLOCK (ID BIGINT NOT NULL, MAPPING LONGBLOB, PRIMARY KEY (ID)); CREATE TABLE MC_SEQUENTIAL_BLOCK (ID BIGINT NOT NULL, PRIMARY KEY (ID)); CREATE TABLE MC_CONCURRENT_BLOCK (ID BIGINT NOT NULL, PRIMARY KEY (ID)); CREATE TABLE MC_SCRIPT_BLOCK (ID BIGINT NOT NULL, SCRIPT VARCHAR(255), PRIMARY KEY (ID)); CREATE TABLE MC_FOLDER_ITEM (ID BIGINT AUTO_INCREMENT NOT NULL, DTYPE VARCHAR(31), FOLDER BIGINT NOT NULL, PRIMARY KEY (ID)); CREATE TABLE MC_ROBOT (ID BIGINT NOT NULL, BYTESIZE INTEGER NOT NULL, CONTAINSRETURNOBJECT TINYINT(1) default 0, CREATEDBY VARCHAR(255), LASTMODIFIED DATETIME NOT NULL, MODIFIEDBY VARCHAR(255), NAME VARCHAR(255) NOT NULL, ROBOTBYTES LONGBLOB NOT NULL, SHA_HASH VARCHAR(255) NOT NULL, VERSION VARCHAR(255) NOT NULL, FOLDER_EX BIGINT NOT NULL, P_ID BIGINT NOT NULL, PRIMARY KEY (ID)); CREATE TABLE MC_TYPE (ID BIGINT NOT NULL, CREATEDBY VARCHAR(255), LASTMODIFIED DATETIME, MODIFIEDBY VARCHAR(255), NAME VARCHAR(255) NOT NULL, SHA_HASH VARCHAR(255) NOT NULL, TYPEBYTES LONGBLOB NOT NULL, P_ID BIGINT	ALTER TABLE MC_SCHEDULE DROP FOREIGN KEY FK_MC_SCHEDULE_BLOCKJOB_ID; ALTER TABLE MC_SCHEDULE DROP FOREIGN KEY FK_MC_SCHEDULE_CLUSTER_ID; ALTER TABLE MC_SCHEDULE DROP FOREIGN KEY FK_MC_SCHEDULE_P_ID; ALTER TABLE MC_SCHEDULE DROP KEY UNQ_MC_SCHEDULE_0; ALTER TABLE MC_BLOCK DROP FOREIGN KEY FK_MC_BLOCK_P_ID; ALTER TABLE MC_BLOCK DROP KEY UNQ_MC_BLOCK_0; ALTER TABLE MC_BLOCK_REFERENCE DROP FOREIGN KEY FK_MC_BLOCK_REFERENCE_PARENT; ALTER TABLE MC_BLOCK_REFERENCE DROP FOREIGN KEY FK_MC_BLOCK_REFERENCE_CHILD; ALTER TABLE MC_BLOCK_REFERENCE DROP KEY UNQ_MC_BLOCK_REFERENCE_0; ALTER TABLE MC_BLOCK_JOB DROP FOREIGN KEY FK_MC_BLOCK_JOB_BLOCK_ID; ALTER TABLE MC_ROBOT_BLOCK DROP FOREIGN KEY FK_MC_ROBOT_BLOCK_ID; ALTER TABLE MC_ROBOT_BLOCK DROP FOREIGN KEY FK_MC_ROBOT_BLOCK_ROBOT_ID; ALTER TABLE MC_MULTIPLE_ROBOT_BLOCK DROP FOREIGN KEY FK_MC_MULTIPLE_ROBOT_BLOCK_ID; ALTER TABLE MC_CONTAINER_BLOCK DROP FOREIGN KEY FK_MC_CONTAINER_BLOCK_ID; ALTER TABLE MC_SEQUENTIAL_BLOCK DROP FOREIGN KEY

Database	Create Tables	Drop Tables
	NOT NULL, PRIMARY KEY (ID)); CREATE TABLE MC_SNIPPET (ID BIGINT NOT NULL, CREATEDBY VARCHAR(255), LASTMODIFIED DATETIME NOT NULL, MODIFIEDBY VARCHAR(255), NAME VARCHAR(255) NOT NULL, SHA_HASH VARCHAR(255) NOT NULL, SNIPPETBYTES LONGBLOB NOT NULL, VERSION VARCHAR(255) NOT NULL, P_ID BIGINT NOT NULL, PRIMARY KEY (ID)); CREATE TABLE MC_ROBOSERVER (ID BIGINT AUTO_INCREMENT NOT NULL, HOST VARCHAR(255) NOT NULL, PORT INTEGER NOT NULL, CLUSTER_ID BIGINT NOT NULL, PRIMARY KEY (ID)); CREATE TABLE MC_SETTINGS (NAME VARCHAR(255) NOT NULL, VALUE VARCHAR(255), PRIMARY KEY (NAME)); CREATE TABLE MC_ROBOT_PARAMETER (ID BIGINT AUTO_INCREMENT NOT NULL, TYPENAME VARCHAR(255) NOT NULL, VARIABLENAME VARCHAR(255) NOT NULL, ROBOT_ID BIGINT NOT NULL, PRIMARY KEY (ID)); CREATE TABLE MC_ROBOT_REF_TYPE (ID BIGINT AUTO_INCREMENT NOT NULL, ISINPUT TINYINT(1) default 0 NOT NULL, ISRETURNED TINYINT(1) default 0 NOT NULL, ISSTORED TINYINT(1) default 0 NOT NULL, NAME VARCHAR(255) NOT NULL, ROBOT_ID BIGINT NOT NULL, PRIMARY KEY (ID)); CREATE TABLE MC_ROBOT_REF_SNIPPET (ID BIGINT AUTO_INCREMENT NOT NULL, NAME VARCHAR(255) NOT NULL, ROBOT_ID BIGINT NOT NULL, PRIMARY KEY (ID)); CREATE TABLE MC_ROBOT_TYPED_VARIABLE (ID BIGINT AUTO_INCREMENT NOT NULL, TYPENAME VARCHAR(255) NOT NULL, VARIABLENAME VARCHAR(255) NOT NULL, ROBOT_ID BIGINT NOT NULL, PRIMARY KEY (ID)); CREATE TABLE MC_ROBOT_REF_VAR (ID BIGINT AUTO_INCREMENT NOT NULL, ISRETURNED TINYINT(1) default 0 NOT NULL, ISSTORED TINYINT(1) default 0 NOT NULL, NAME VARCHAR(255), ROBOT_ID BIGINT NOT NULL, PRIMARY KEY (ID)); CREATE TABLE MC_RESOURCE_FILE (ID BIGINT NOT NULL, BYTESIZE INTEGER NOT NULL, BYTES LONGBLOB NOT NULL, CREATEDBY VARCHAR(255), LASTMODIFIED DATETIME NOT NULL, MODIFIEDBY VARCHAR(255), NAME VARCHAR(255) NOT NULL, SHA_HASH VARCHAR(255) NOT NULL, FOLDER_EX BIGINT NOT NULL, P_ID BIGINT NOT NULL, PRIMARY KEY (ID)); CREATE TABLE MC_PROJECT (ID BIGINT AUTO_INCREMENT NOT NULL, ACAO VARCHAR(255), AUTHENTICATE TINYINT(1) default 0, DESCRIPTION VARCHAR(255), NAME VARCHAR(255) NOT NULL UNIQUE, CLUSTER_ID BIGINT, PRIMARY KEY (ID)); CREATE TABLE MC_ROBOSERVER_CLUSTER (ID BIGINT AUTO_INCREMENT NOT NULL, KCU INTEGER, NAME VARCHAR(255) NOT NULL UNIQUE, type TINYINT(1) default 0 NOT NULL, USESSL TINYINT(1) default 0 NOT NULL, SETTINGS_ID BIGINT, PRIMARY KEY (ID));	FK_MC_SEQUENTIAL_BLOCK_ID; ALTER TABLE MC_CONCURRENT_BLOCK DROP FOREIGN KEY FK_MC_CONCURRENT_BLOCK_ID; ALTER TABLE MC_SCRIPT_BLOCK DROP FOREIGN KEY FK_MC_SCRIPT_BLOCK_ID; ALTER TABLE MC_FOLDER_ITEM DROP FOREIGN KEY FK_MC_FOLDER_ITEM_FOLDER; ALTER TABLE MC_ROBOT DROP FOREIGN KEY FK_MC_ROBOT_FOLDER_EX; ALTER TABLE MC_ROBOT DROP FOREIGN KEY FK_MC_ROBOT_ID; ALTER TABLE MC_ROBOT DROP FOREIGN KEY FK_MC_ROBOT_P_ID; ALTER TABLE MC_ROBOT DROP KEY UNQ_MC_ROBOT_0; ALTER TABLE MC_TYPE DROP FOREIGN KEY FK_MC_TYPE_P_ID; ALTER TABLE MC_TYPE DROP FOREIGN KEY FK_MC_TYPE_ID; ALTER TABLE MC_TYPE DROP KEY UNQ_MC_TYPE_0; ALTER TABLE MC_SNIPPET DROP FOREIGN KEY FK_MC_SNIPPET_P_ID; ALTER TABLE MC_SNIPPET DROP FOREIGN KEY FK_MC_SNIPPET_ID; ALTER TABLE MC_SNIPPET DROP KEY UNQ_MC_SNIPPET_0; ALTER TABLE MC_ROBOSERVER DROP FOREIGN KEY FK_MC_ROBOSERVER_CLUSTER_ID; ALTER TABLE MC_ROBOT_PARAMETER DROP FOREIGN KEY

Database	Create Tables	Drop Tables
	CREATE TABLE MC_GROUP_TO_ROLE_MAP (ID BIGINT AUTO_INCREMENT NOT NULL, GROUP_NAME VARCHAR(200) NOT NULL, ROLE_NAME VARCHAR(40) NOT NULL, P_ID BIGINT NOT NULL, PRIMARY KEY (ID)); CREATE TABLE MC_SNIPPET_PARAMETER (ID BIGINT AUTO_INCREMENT NOT NULL, TYPENAME VARCHAR(255) NOT NULL, VARIABLENAME VARCHAR(255) NOT NULL, SNIPPET_ID BIGINT NOT NULL, PRIMARY KEY (ID)); CREATE TABLE MC_SNIPPET_REF_SNIPPET (ID BIGINT AUTO_INCREMENT NOT NULL, NAME VARCHAR(255) NOT NULL, SNIPPET_ID BIGINT NOT NULL, PRIMARY KEY (ID)); CREATE TABLE MC_SNIPPET_REF_TYPE (ID BIGINT AUTO_INCREMENT NOT NULL, ISINPUT TINYINT(1) default 0 NOT NULL, ISRETURNED TINYINT(1) default 0 NOT NULL, ISSTORED TINYINT(1) default 0 NOT NULL, NAME VARCHAR(255), SNIPPET_ID BIGINT NOT NULL, PRIMARY KEY (ID)); CREATE TABLE MC_SNIPPET_TYPED_VAR (ID BIGINT AUTO_INCREMENT NOT NULL, TYPENAME VARCHAR(255) NOT NULL, VARIABLENAME VARCHAR(255) NOT NULL, SNIPPET_ID BIGINT NOT NULL, PRIMARY KEY (ID)); CREATE TABLE MC_SNIPPET_REF_VAR (ID BIGINT AUTO_INCREMENT NOT NULL, ISRETURNED TINYINT(1) default 0 NOT NULL, ISSTORED TINYINT(1) default 0 NOT NULL, NAME VARCHAR(255), SNIPPET_ID BIGINT NOT NULL, PRIMARY KEY (ID)); CREATE TABLE MC_OAUTH_APPLICATION (ID BIGINT AUTO_INCREMENT NOT NULL, CALLBACKURL VARCHAR(255), CONSUMERKEY VARCHAR(255), CONSUMERSECRET LONGTEXT, NAME VARCHAR(255) NOT NULL, SCOPE VARCHAR(255), SERVICEPROVIDER VARCHAR(255), P_ID BIGINT NOT NULL, PRIMARY KEY (ID)); CREATE TABLE MC_OAUTH_USER (ID BIGINT NOT NULL, ACCESSTOKEN VARCHAR(255), ACCESSTOKENSECRET LONGTEXT, NAME VARCHAR(255) NOT NULL, REFRESHTOKEN VARCHAR(255), APPLICATION_ID BIGINT NOT NULL, USER_ID VARCHAR(255), PRIMARY KEY (ID)); CREATE TABLE MC_DSLICENSE (TOKEN VARCHAR(255) NOT NULL, EXPIRATION BIGINT, IP VARCHAR(255), USERNAME VARCHAR(255), PRIMARY KEY (TOKEN)); CREATE TABLE MC_USER (USER_NAME VARCHAR(255) NOT NULL, ADMIN_USER TINYINT(1) default 0, CREATION_DATE DATETIME, EMAIL_ADDR VARCHAR(255), FULL_NAME VARCHAR(255), LAST_IP_ADDR VARCHAR(255), LOGIN_TIME DATETIME, LOGIN_COUNT INTEGER NOT NULL, PASS_WORD VARCHAR(255), PRIMARY KEY (USER_NAME)); CREATE TABLE MC_GROUP (ID BIGINT AUTO_INCREMENT NOT NULL, DESCRIPTION VARCHAR(255), GROUP_NAME VARCHAR(255) NOT NULL, PRIMARY KEY (ID)); CREATE TABLE MC_USER_GROUP_REL (ID BIGINT AUTO_INCREMENT NOT NULL, GROUP_ID BIGINT NOT NULL, USER_ID	FK_MC_ROBOT_PARAMETER_ROBOT_ID; ALTER TABLE MC_ROBOT_REF_TYPE DROP FOREIGN KEY FK_MC_ROBOT_REF_TYPE_ROBOT_ID; ALTER TABLE MC_ROBOT_REF_SNIPPET DROP FOREIGN KEY FK_MC_ROBOT_REF_SNIPPET_ROBOT_I ALTER TABLE MC_ROBOT_TYPED_VARIABLE DROP FOREIGN KEY FK_MC_ROBOT_TYPED_VARIABLE_ROBO ALTER TABLE MC_ROBOT_REF_VAR DROP FOREIGN KEY FK_MC_ROBOT_REF_VAR_ROBOT_ID; ALTER TABLE MC_RESOURCE_FILE DROP FOREIGN KEY FK_MC_RESOURCE_FILE_FOLDER_EX; ALTER TABLE MC_RESOURCE_FILE DROP FOREIGN KEY FK_MC_RESOURCE_FILE_ID; ALTER TABLE MC_RESOURCE_FILE DROP FOREIGN KEY FK_MC_RESOURCE_FILE_P_ID; ALTER TABLE MC_RESOURCE_FILE DROP KEY UNQ_MC_RESOURCE_FILE_0; ALTER TABLE MC_PROJECT DROP FOREIGN KEY FK_MC_PROJECT_CLUSTER_ID; ALTER TABLE MC_ROBOSERVER_CLUSTER DROP FOREIGN KEY FK_MC_ROBOSERVER_CLUSTER_SETTIN ALTER TABLE MC_GROUP_TO_ROLE_MAP DROP FOREIGN KEY FK_MC_GROUP_TO_ROLE_MAP_P_ID; ALTER TABLE MC_GROUP_TO_ROLE_MAP DROP KEY UNQ_MC_GROUP_TO_ROLE_MAP_0; ALTER TABLE MC_SNIPPET_PARAMETER DROP FOREIGN KEY FK_MC_SNIPPET_PARAMETER_SNIPPET ALTER TABLE MC_SNIPPET_REF_SNIPPET DROP FOREIGN KEY FK_MC_SNIPPET_REF_SNIPPET_SNIPP ALTER TABLE

Database	Create Tables	Drop Tables
	VARCHAR(255) NOT NULL, PRIMARY KEY (ID)); CREATE TABLE MC_USER_PROP (ID BIGINT AUTO_INCREMENT NOT NULL, IS_NULL TINYINT(1) default 0, PROP_NAME VARCHAR(255) NOT NULL, PROP_VALUE LONGTEXT, USER_ID VARCHAR(255) NOT NULL, PRIMARY KEY (ID)); CREATE TABLE MC_TYPE_ATTR (ID BIGINT AUTO_INCREMENT NOT NULL, ATTR_TYPE VARCHAR(255), COMMENTS LONGTEXT, DEFAULT_VALUE LONGTEXT, NAME VARCHAR(255), REQUIRED_ATTR TINYINT(1) default 0, TYPE_ORDER INTEGER, TYPE_ID BIGINT NOT NULL, PRIMARY KEY (ID)); CREATE TABLE MC_RS_CLUSTER_SETTINGS (ID BIGINT AUTO_INCREMENT NOT NULL, CLUSTER_SETTINGS_ID VARCHAR(255) NOT NULL, EMAIL_LOG_FROM_ADDRESS VARCHAR(255), LOG_INPUT_TO_DB TINYINT(1) default 0, LOG_INPUT_TO_LOG4J TINYINT(1) default 0, LOG_TO_DB TINYINT(1) default 0 NOT NULL, LOG_TO_EMAIL TINYINT(1) default 0 NOT NULL, LOG_TO_LOG4J TINYINT(1) default 0 NOT NULL, MAX_CONCURRENT_ROBOTS INTEGER, MAX_QUEUED_ROBOTS INTEGER, PROF_ENABLED TINYINT(1) default 0, PROF_FILE_OUTPUT_APPEND TINYINT(1) default 0, PROF_FILE_OUTPUT_FILE VARCHAR(255), PROF_LOG_PAGE_URL TINYINT(1) default 0, PROF_OUTPUT_TARGET VARCHAR(20), PROF_TYPE VARCHAR(20), PROF_THRESHOLD INTEGER, EMAIL_LOG_TO_ADDRESS VARCHAR(255), PRIMARY KEY (ID)); CREATE TABLE MC_DATABASE (ID BIGINT AUTO_INCREMENT NOT NULL, HOST VARCHAR(255) NOT NULL, MAX_ACTIVE_CONNECTIONS INTEGER, MAX_IDLE_CONNECTIONS INTEGER, NAME VARCHAR(255) NOT NULL, PASSWORD VARCHAR(255), SCHEMA_NAME VARCHAR(255) NOT NULL, TYPE VARCHAR(255) NOT NULL, USERNAME VARCHAR(255), CLUSTER_SETTINGS_ID BIGINT NOT NULL, PRIMARY KEY (ID)); CREATE TABLE MC_PROXY_SERVER (ID BIGINT AUTO_INCREMENT NOT NULL, EXCLUDED_HOST_NAMES VARCHAR(1000), HOST_NAME VARCHAR(255) NOT NULL, PASSWORD VARCHAR(255), PORT_NUMBER INTEGER, USERNAME VARCHAR(255), CLUSTER_SETTINGS_ID BIGINT NOT NULL, PRIMARY KEY (ID)); CREATE TABLE MC_DATABASE_TYPE (ID BIGINT AUTO_INCREMENT NOT NULL, DRIVER_CLASS VARCHAR(255) NOT NULL, NAME VARCHAR(255) NOT NULL, URL_TEMPLATE VARCHAR(255) NOT NULL, VALIDATION_QUERY VARCHAR(255) NOT NULL, PRIMARY KEY (ID)); CREATE TABLE MC_JAR_FILE (ID BIGINT AUTO_INCREMENT NOT NULL, BYTES LONGBLOB, CREATED DATETIME, NAME VARCHAR(255) NOT NULL, FILE_SIZE INTEGER, PRIMARY KEY (ID)); CREATE TABLE APP_KAPPLET (ID VARCHAR(40) NOT NULL, DTYPE VARCHAR(31), CREATED_DATE DATETIME	MC_SNIPPET_REF_TYPE DROP FOREIGN KEY FK_MC_SNIPPET_REF_TYPE_SNIPPET_ ALTER TABLE MC_SNIPPET_TYPED_VAR DROP FOREIGN KEY FK_MC_SNIPPET_TYPED_VAR_SNIPPET ALTER TABLE MC_SNIPPET_REF_VAR DROP FOREIGN KEY FK_MC_SNIPPET_REF_VAR_SNIPPET_I ALTER TABLE MC_OAUTH_APPLICATION DROP FOREIGN KEY FK_MC_OAUTH_APPLICATION_P_ID; ALTER TABLE MC_OAUTH_APPLICATION DROP KEY UNQ_MC_OAUTH_APPLICATION_0; ALTER TABLE MC_OAUTH_USER DROP FOREIGN KEY FK_MC_OAUTH_USER_APPLICATION_ID ALTER TABLE MC_OAUTH_USER DROP FOREIGN KEY FK_MC_OAUTH_USER_USER_ID; ALTER TABLE MC_OAUTH_USER DROP KEY UNQ_MC_OAUTH_USER_0; ALTER TABLE MC_GROUP DROP KEY UNQ_MC_GROUP_0; ALTER TABLE MC_USER_GROUP_REL DROP FOREIGN KEY FK_MC_USER_GROUP_REL_GROUP_ID; ALTER TABLE MC_USER_GROUP_REL DROP FOREIGN KEY FK_MC_USER_GROUP_REL_USER_ID; ALTER TABLE MC_USER_GROUP_REL DROP KEY UNQ_MC_USER_GROUP_REL_0; ALTER TABLE MC_USER_PROP DROP FOREIGN KEY FK_MC_USER_PROP_USER_ID; ALTER TABLE MC_USER_PROP DROP KEY UNQ_MC_USER_PROP_0; ALTER TABLE MC_TYPE_ATTR DROP FOREIGN KEY FK_MC_TYPE_ATTR_TYPE_ID;

Database	Create Tables	Drop Tables
	NOT NULL, DESCRIPTION LONGTEXT, NAME VARCHAR(255) NOT NULL, OPTIONS LONGTEXT, P_ID BIGINT NOT NULL, ICON_ID VARCHAR(255), PRIMARY KEY (ID)); CREATE TABLE APP_MASTER (ID VARCHAR(40) NOT NULL, CREATEDBY VARCHAR(255), ENABLED TINYINT(1) default 0 NOT NULL, FLOW LONGTEXT, INITIAL_LABEL INTEGER, KAPPLET_TYPE VARCHAR(255) NOT NULL, LAST_INSTALLED DATETIME, LAST_MODIFIED DATETIME NOT NULL, LAST_RUN DATETIME, MODIFIEDBY VARCHAR(255), PRIMARY KEY (ID)); CREATE TABLE APP_INSTALLED (ID VARCHAR(40) NOT NULL, FAVORITE TINYINT(1) default 0, LAST_RUN DATETIME, UPDATED_AT DATETIME, PARENT VARCHAR(40) NOT NULL, USER_ID VARCHAR(255) NOT NULL, PRIMARY KEY (ID)); CREATE TABLE APP_ICON (ID VARCHAR(255) NOT NULL, ICON LONGBLOB, filename VARCHAR(255), mimeType VARCHAR(255), PRIMARY KEY (ID)); CREATE TABLE APP_UPLOAD (ID VARCHAR(255) NOT NULL, UPLOAD LONGBLOB NOT NULL, filename VARCHAR(255) NOT NULL, mimeType VARCHAR(255) NOT NULL, timestamp DATETIME NOT NULL, username VARCHAR(255) NOT NULL, PRIMARY KEY (ID)); CREATE TABLE MC_KAPPLET_PROP (ID BIGINT AUTO_INCREMENT NOT NULL, IS_NULL TINYINT(1) default 0, PROP_NAME VARCHAR(255) NOT NULL, PROP_VALUE LONGTEXT, KAPPLET_ID VARCHAR(40) NOT NULL, PRIMARY KEY (ID)); CREATE TABLE APP_SCHEDULE_ERROR (ID BIGINT AUTO_INCREMENT NOT NULL, ERROR LONGTEXT, SCHEDULE_ID BIGINT NOT NULL, PRIMARY KEY (ID)); CREATE TABLE APP_SCHEDULE (ID BIGINT NOT NULL, DIRTY TINYINT(1) default 0, LASTRUN DATETIME, NEXTRUN DATETIME, PARAMETERS LONGTEXT, APP_ID VARCHAR(40) NOT NULL, PRIMARY KEY (ID)); CREATE TABLE APP_SCHEDULE_VALUE (ID VARCHAR(255) NOT NULL, VALUE LONGBLOB NOT NULL, SCHEDULE_ID BIGINT NOT NULL, PRIMARY KEY (ID)); CREATE TABLE MC_GLOBAL_KAPPLET_PROP (PROP_NAME VARCHAR(255) NOT NULL, IS_NULL TINYINT(1) default 0, PROP_VALUE LONGTEXT, PRIMARY KEY (PROP_NAME)); CREATE TABLE APP_ACTION_EXECUTION (ID BIGINT NOT NULL, END_TIME DATETIME, LABEL INTEGER NOT NULL, SEQ_ID INTEGER, START_TIME DATETIME, STATUS VARCHAR(255), TASK_IDS VARCHAR(255), VERSIONLOCK BIGINT, KAPPLET_RUN_ID BIGINT NOT NULL, PRIMARY KEY (ID)); CREATE TABLE APP_DATA_ROW (ID BIGINT NOT NULL, TABLE_ID BIGINT NOT NULL, PRIMARY KEY (ID)); CREATE TABLE APP_DATA_TABLE (ID BIGINT NOT NULL, TABLE_NAME VARCHAR(255) NOT NULL, FLOW_EXECUTION_ID BIGINT NOT NULL, PRIMARY KEY (ID)); CREATE TABLE APP_DATA_STR (ID BIGINT NOT NULL, NAME VARCHAR(255), VALUE VARCHAR(2000) NOT NULL, RESULT_ID BIGINT	ALTER TABLE MC_DATABASE DROP FOREIGN KEY FK_MC_DATABASE_CLUSTER_SETTINGS ALTER TABLE MC_PROXY_SERVER DROP FOREIGN KEY FK_MC_PROXY_SERVER_CLUSTER_SETT ALTER TABLE APP_KAPPLET DROP FOREIGN KEY FK_APP_KAPPLET_ICON_ID; ALTER TABLE APP_KAPPLET DROP FOREIGN KEY FK_APP_KAPPLET_P_ID; ALTER TABLE APP_MASTER DROP FOREIGN KEY FK_APP_MASTER_ID; ALTER TABLE APP_INSTALLED DROP FOREIGN KEY FK_APP_INSTALLED_USER_ID; ALTER TABLE APP_INSTALLED DROP FOREIGN KEY FK_APP_INSTALLED_PARENT; ALTER TABLE APP_INSTALLED DROP FOREIGN KEY FK_APP_INSTALLED_ID; ALTER TABLE MC_KAPPLET_PROP DROP FOREIGN KEY FK_MC_KAPPLET_PROP_KAPPLET_ID; ALTER TABLE MC_KAPPLET_PROP DROP KEY UNQ_MC_KAPPLET_PROP_0; ALTER TABLE APP_SCHEDULE_ERROR DROP FOREIGN KEY FK_APP_SCHEDULE_ERROR_SCHEDULE_ ALTER TABLE APP_SCHEDULE DROP FOREIGN KEY FK_APP_SCHEDULE_APP_ID; ALTER TABLE APP_SCHEDULE_VALUE DROP FOREIGN KEY FK_APP_SCHEDULE_VALUE_SCHEDULE_ ALTER TABLE APP_ACTION_EXECUTION DROP FOREIGN KEY FK_APP_ACTION_EXECUTION_KAPPLET ALTER TABLE APP_ACTION_EXECUTION

Database	Create Tables	Drop Tables
	NOT NULL, PRIMARY KEY (ID)); CREATE TABLE APP_DATA_LOB (ID BIGINT NOT NULL, LARGE TINYINT(1) default 0, NAME VARCHAR(255), VALUE LONGTEXT, RESULT_ID BIGINT NOT NULL, PRIMARY KEY (ID)); CREATE TABLE APP_FLOW_ERROR (ID BIGINT AUTO_INCREMENT NOT NULL, ERROR LONGTEXT, EXECUTION_ID BIGINT NOT NULL, PRIMARY KEY (ID)); CREATE TABLE APP_RUN (ID BIGINT AUTO_INCREMENT NOT NULL, ABORTED TINYINT(1) default 0, FLOW LONGTEXT, INITIAL_LABEL INTEGER NOT NULL, LASTEXECUTIONPATHCHANGE BIGINT, OPTIONS LONGTEXT, PARAMETERS LONGTEXT, SIGNATURES LONGTEXT, VERSIONLOCK BIGINT, KAPPLET_ID VARCHAR(40) NOT NULL, PRIMARY KEY (ID)); CREATE TABLE APP_RUN_VALUE (ID VARCHAR(255) NOT NULL, VALUE LONGBLOB NOT NULL, RUN_ID BIGINT NOT NULL, PRIMARY KEY (ID)); CREATE TABLE MC_FOLDER (ID BIGINT AUTO_INCREMENT NOT NULL, DTYPE VARCHAR(31), PATH VARCHAR(255), PRIMARY KEY (ID)); CREATE TABLE MC_ROOT_FOLDER (ID BIGINT NOT NULL, P_ID BIGINT NOT NULL, PRIMARY KEY (ID)); CREATE TABLE MC_SUBFOLDER (ID BIGINT NOT NULL, NAME VARCHAR(255) NOT NULL, PARENT_ID BIGINT NOT NULL, PRIMARY KEY (ID)); CREATE TABLE APP_RUN_APP_SCHEDULE (schedules_ID BIGINT NOT NULL, kappletRuns_ID BIGINT NOT NULL, PRIMARY KEY (schedules_ID, kappletRuns_ID)); ALTER TABLE MC_SCHEDULE ADD CONSTRAINT UNQ_MC_SCHEDULE_0 UNIQUE (P_ID, NAME); ALTER TABLE MC_BLOCK ADD CONSTRAINT UNQ_MC_BLOCK_0 UNIQUE (P_ID, NAME); ALTER TABLE MC_BLOCK_REFERENCE ADD CONSTRAINT UNQ_MC_BLOCK_REFERENCE_0 UNIQUE (PARENT, NAME); ALTER TABLE MC_ROBOT ADD CONSTRAINT UNQ_MC_ROBOT_0 UNIQUE (P_ID, NAME, FOLDER_EX); ALTER TABLE MC_TYPE ADD CONSTRAINT UNQ_MC_TYPE_0 UNIQUE (P_ID, NAME); ALTER TABLE MC_SNIPPET ADD CONSTRAINT UNQ_MC_SNIPPET_0 UNIQUE (P_ID, NAME); ALTER TABLE MC_RESOURCE_FILE ADD CONSTRAINT UNQ_MC_RESOURCE_FILE_0 UNIQUE (P_ID, NAME, FOLDER_EX); ALTER TABLE MC_GROUP_TO_ROLE_MAP ADD CONSTRAINT UNQ_MC_GROUP_TO_ROLE_MAP_0 UNIQUE (P_ID, ROLE_NAME, GROUP_NAME); ALTER TABLE MC_OAUTH_APPLICATION ADD CONSTRAINT UNQ_MC_OAUTH_APPLICATION_0 UNIQUE (P_ID, NAME); ALTER TABLE MC_OAUTH_USER ADD CONSTRAINT UNQ_MC_OAUTH_USER_0 UNIQUE (APPLICATION_ID, NAME); ALTER TABLE MC_GROUP ADD CONSTRAINT UNQ_MC_GROUP_0 UNIQUE (GROUP_NAME); ALTER TABLE MC_USER_GROUP_REL ADD CONSTRAINT UNQ_MC_USER_GROUP_REL_0 UNIQUE (USER_ID, GROUP_ID); ALTER TABLE MC_USER_PROP ADD CONSTRAINT UNQ_MC_USER_PROP_0 UNIQUE (ID, PROP_NAME); ALTER TABLE MC_KAPPLET_PROP ADD CONSTRAINT	DROP KEY UNQ_APP_ACTION_EXECUTION_0; ALTER TABLE APP_DATA_ROW DROP FOREIGN KEY FK_APP_DATA_ROW_TABLE_ID; ALTER TABLE APP_DATA_TABLE DROP FOREIGN KEY FK_APP_DATA_TABLE_FLOW_EXECUTIO ALTER TABLE APP_DATA_TABLE DROP KEY UNQ_APP_DATA_TABLE_0; ALTER TABLE APP_DATA_STR DROP FOREIGN KEY FK_APP_DATA_STR_RESULT_ID; ALTER TABLE APP_DATA_LOB DROP FOREIGN KEY FK_APP_DATA_LOB_RESULT_ID; ALTER TABLE APP_FLOW_ERROR DROP FOREIGN KEY FK_APP_FLOW_ERROR_EXECUTION_ID; ALTER TABLE APP_RUN DROP FOREIGN KEY FK_APP_RUN_KAPPLET_ID; ALTER TABLE APP_RUN_VALUE DROP FOREIGN KEY FK_APP_RUN_VALUE_RUN_ID; ALTER TABLE MC_ROOT_FOLDER DROP FOREIGN KEY FK_MC_ROOT_FOLDER_P_ID; ALTER TABLE MC_ROOT_FOLDER DROP FOREIGN KEY FK_MC_ROOT_FOLDER_ID; ALTER TABLE MC_SUBFOLDER DROP FOREIGN KEY FK_MC_SUBFOLDER_ID; ALTER TABLE MC_SUBFOLDER DROP FOREIGN KEY FK_MC_SUBFOLDER_PARENT_ID; ALTER TABLE MC_SUBFOLDER DROP KEY UNQ_MC_SUBFOLDER_0; ALTER TABLE APP_RUN_APP_SCHEDULE DROP FOREIGN KEY FK_APP_RUN_APP_SCHEDULE_kapplet

Database	Create Tables	Drop Tables
	UNQ_MC_KAPPLET_PROP_0 UNIQUE (ID, PROP_NAME); ALTER TABLE APP_ACTION_EXECUTION ADD CONSTRAINT UNQ_APP_ACTION_EXECUTION_0 UNIQUE (KAPPLET_RUN_ID, LABEL); ALTER TABLE APP_DATA_TABLE ADD CONSTRAINT UNQ_APP_DATA_TABLE_0 UNIQUE (FLOW_EXECUTION_ID, TABLE_NAME); ALTER TABLE MC_SUBFOLDER ADD CONSTRAINT UNQ_MC_SUBFOLDER_0 UNIQUE (NAME, PARENT_ID); ALTER TABLE MC_SCHEDULE ADD CONSTRAINT FK_MC_SCHEDULE_BLOCKJOB_ID FOREIGN KEY (BLOCKJOB_ID) REFERENCES MC_BLOCK_JOB (ID); ALTER TABLE MC_SCHEDULE ADD CONSTRAINT FK_MC_SCHEDULE_CLUSTER_ID FOREIGN KEY (CLUSTER_ID) REFERENCES MC_ROBOSERVER_CLUSTER (ID); ALTER TABLE MC_SCHEDULE ADD CONSTRAINT FK_MC_SCHEDULE_P_ID FOREIGN KEY (P_ID) REFERENCES MC_PROJECT (ID); ALTER TABLE MC_BLOCK ADD CONSTRAINT FK_MC_BLOCK_P_ID FOREIGN KEY (P_ID) REFERENCES MC_PROJECT (ID); ALTER TABLE MC_BLOCK_REFERENCE ADD CONSTRAINT FK_MC_BLOCK_REFERENCE_PARENT FOREIGN KEY (PARENT) REFERENCES MC_BLOCK (ID); ALTER TABLE MC_BLOCK_REFERENCE ADD CONSTRAINT FK_MC_BLOCK_REFERENCE_CHILD FOREIGN KEY (CHILD) REFERENCES MC_BLOCK (ID); ALTER TABLE MC_BLOCK_JOB ADD CONSTRAINT FK_MC_BLOCK_JOB_BLOCK_ID FOREIGN KEY (BLOCK_ID) REFERENCES MC_BLOCK (ID); ALTER TABLE MC_ROBOT_BLOCK ADD CONSTRAINT FK_MC_ROBOT_BLOCK_ID FOREIGN KEY (ID) REFERENCES MC_BLOCK (ID); ALTER TABLE MC_ROBOT_BLOCK ADD CONSTRAINT FK_MC_ROBOT_BLOCK_ROBOT_ID FOREIGN KEY (ROBOT_ID) REFERENCES MC_FOLDER_ITEM (ID); ALTER TABLE MC_MULTIPLE_ROBOT_BLOCK ADD CONSTRAINT FK_MC_MULTIPLE_ROBOT_BLOCK_ID FOREIGN KEY (ID) REFERENCES MC_BLOCK (ID); ALTER TABLE MC_CONTAINER_BLOCK ADD CONSTRAINT FK_MC_CONTAINER_BLOCK_ID FOREIGN KEY (ID) REFERENCES MC_BLOCK (ID); ALTER TABLE MC_SEQUENTIAL_BLOCK ADD CONSTRAINT FK_MC_SEQUENTIAL_BLOCK_ID FOREIGN KEY (ID) REFERENCES MC_BLOCK (ID); ALTER TABLE MC_CONCURRENT_BLOCK ADD CONSTRAINT FK_MC_CONCURRENT_BLOCK_ID FOREIGN KEY (ID) REFERENCES MC_BLOCK (ID); ALTER TABLE MC_SCRIPT_BLOCK ADD CONSTRAINT FK_MC_SCRIPT_BLOCK_ID FOREIGN KEY (ID) REFERENCES MC_BLOCK (ID); ALTER TABLE MC_FOLDER_ITEM ADD CONSTRAINT FK_MC_FOLDER_ITEM_FOLDER FOREIGN KEY (FOLDER) REFERENCES MC_FOLDER (ID); ALTER TABLE MC_ROBOT ADD CONSTRAINT FK_MC_ROBOT_FOLDER_EX FOREIGN KEY (FOLDER_EX) REFERENCES MC_FOLDER (ID); ALTER TABLE MC_ROBOT ADD CONSTRAINT	ALTER TABLE APP_RUN_APP_SCHEDULE DROP FOREIGN KEY FK_APP_RUN_APP_SCHEDULE_schedul DROP TABLE MC_SCHEDULE; DROP TABLE MC_BLOCK; DROP TABLE MC_BLOCK_REFERENCE; DROP TABLE MC_BLOCK_JOB; DROP TABLE MC_ROBOT_BLOCK; DROP TABLE MC_MULTIPLE_ROBOT_BLOCK; DROP TABLE MC_CONTAINER_BLOCK; DROP TABLE MC_SEQUENTIAL_BLOCK; DROP TABLE MC_CONCURRENT_BLOCK; DROP TABLE MC_SCRIPT_BLOCK; DROP TABLE MC_FOLDER_ITEM; DROP TABLE MC_ROBOT; DROP TABLE MC_TYPE; DROP TABLE MC_SNIPPET; DROP TABLE MC_ROBOSERVER; DROP TABLE MC_SETTINGS; DROP TABLE MC_ROBOT_PARAMETER; DROP TABLE MC_ROBOT_REF_TYPE; DROP TABLE MC_ROBOT_REF_SNIPPET; DROP TABLE MC_ROBOT_TYPED_VARIABLE; DROP TABLE MC_ROBOT_REF_VAR; DROP TABLE MC_RESOURCE_FILE; DROP TABLE MC_PROJECT; DROP TABLE MC_ROBOSERVER_CLUSTER; DROP TABLE MC_GROUP_TO_ROLE_MAP; DROP TABLE MC_SNIPPET_PARAMETER; DROP TABLE MC_SNIPPET_REF_SNIPPET; DROP TABLE MC_SNIPPET_REF_TYPE;

Database	Create Tables	Drop Tables
	FK_MC_ROBOT_ID FOREIGN KEY (ID) REFERENCES MC_FOLDER_ITEM (ID); ALTER TABLE MC_ROBOT ADD CONSTRAINT FK_MC_ROBOT_P_ID FOREIGN KEY (P_ID) REFERENCES MC_PROJECT (ID); ALTER TABLE MC_TYPE ADD CONSTRAINT FK_MC_TYPE_P_ID FOREIGN KEY (P_ID) REFERENCES MC_PROJECT (ID); ALTER TABLE MC_TYPE ADD CONSTRAINT FK_MC_TYPE_ID FOREIGN KEY (ID) REFERENCES MC_FOLDER_ITEM (ID); ALTER TABLE MC_SNIPPET ADD CONSTRAINT FK_MC_SNIPPET_P_ID FOREIGN KEY (P_ID) REFERENCES MC_PROJECT (ID); ALTER TABLE MC_SNIPPET ADD CONSTRAINT FK_MC_SNIPPET_ID FOREIGN KEY (ID) REFERENCES MC_FOLDER_ITEM (ID); ALTER TABLE MC_ROBOSERVER ADD CONSTRAINT FK_MC_ROBOSERVER_CLUSTER_ID FOREIGN KEY (CLUSTER_ID) REFERENCES MC_ROBOSERVER_CLUSTER (ID); ALTER TABLE MC_ROBOT_PARAMETER ADD CONSTRAINT FK_MC_ROBOT_PARAMETER_ROBOT_ID FOREIGN KEY (ROBOT_ID) REFERENCES MC_FOLDER_ITEM (ID); ALTER TABLE MC_ROBOT_REF_TYPE ADD CONSTRAINT FK_MC_ROBOT_REF_TYPE_ROBOT_ID FOREIGN KEY (ROBOT_ID) REFERENCES MC_FOLDER_ITEM (ID); ALTER TABLE MC_ROBOT_REF_SNIPPET ADD CONSTRAINT FK_MC_ROBOT_REF_SNIPPET_ROBOT_ID FOREIGN KEY (ROBOT_ID) REFERENCES MC_FOLDER_ITEM (ID); ALTER TABLE MC_ROBOT_TYPED_VARIABLE ADD CONSTRAINT FK_MC_ROBOT_TYPED_VARIABLE_ROBOT_ID FOREIGN KEY (ROBOT_ID) REFERENCES MC_FOLDER_ITEM (ID); ALTER TABLE MC_ROBOT_REF_VAR ADD CONSTRAINT FK_MC_ROBOT_REF_VAR_ROBOT_ID FOREIGN KEY (ROBOT_ID) REFERENCES MC_FOLDER_ITEM (ID); ALTER TABLE MC_RESOURCE_FILE ADD CONSTRAINT FK_MC_RESOURCE_FILE_FOLDER_EX FOREIGN KEY (FOLDER_EX) REFERENCES MC_FOLDER (ID); ALTER TABLE MC_RESOURCE_FILE ADD CONSTRAINT FK_MC_RESOURCE_FILE_ID FOREIGN KEY (ID) REFERENCES MC_FOLDER_ITEM (ID); ALTER TABLE MC_RESOURCE_FILE ADD CONSTRAINT FK_MC_RESOURCE_FILE_P_ID FOREIGN KEY (P_ID) REFERENCES MC_PROJECT (ID); ALTER TABLE MC_PROJECT ADD CONSTRAINT FK_MC_PROJECT_CLUSTER_ID FOREIGN KEY (CLUSTER_ID) REFERENCES MC_ROBOSERVER_CLUSTER (ID); ALTER TABLE MC_ROBOSERVER_CLUSTER ADD CONSTRAINT FK_MC_ROBOSERVER_CLUSTER_SETTINGS_ID FOREIGN KEY (SETTINGS_ID) REFERENCES MC_RS_CLUSTER_SETTINGS (ID); ALTER TABLE MC_GROUP_TO_ROLE_MAP ADD CONSTRAINT FK_MC_GROUP_TO_ROLE_MAP_P_ID FOREIGN KEY (P_ID) REFERENCES MC_PROJECT (ID); ALTER TABLE MC_SNIPPET_PARAMETER ADD CONSTRAINT FK_MC_SNIPPET_PARAMETER_SNIPPET_ID FOREIGN KEY (SNIPPET_ID) REFERENCES	DROP TABLE MC_SNIPPET_TYPED_VAR; DROP TABLE MC_SNIPPET_REF_VAR; DROP TABLE MC_OAUTH_APPLICATION; DROP TABLE MC_OAUTH_USER; DROP TABLE MC_DSLICENSE; DROP TABLE MC_USER; DROP TABLE MC_GROUP; DROP TABLE MC_USER_GROUP_REL; DROP TABLE MC_USER_PROP; DROP TABLE MC_TYPE_ATTR; DROP TABLE MC_RS_CLUSTER_SETTINGS; DROP TABLE MC_DATABASE; DROP TABLE MC_PROXY_SERVER; DROP TABLE MC_DATABASE_TYPE; DROP TABLE MC_JAR_FILE; DROP TABLE APP_KAPPLET; DROP TABLE APP_MASTER; DROP TABLE APP_INSTALLED; DROP TABLE APP_ICON; DROP TABLE APP_UPLOAD; DROP TABLE MC_KAPPLET_PROP; DROP TABLE APP_SCHEDULE_ERROR; DROP TABLE APP_SCHEDULE; DROP TABLE APP_SCHEDULE_VALUE; DROP TABLE MC_GLOBAL_KAPPLET_PROP; DROP TABLE APP_ACTION_EXECUTION; DROP TABLE APP_DATA_ROW; DROP TABLE APP_DATA_TABLE; DROP TABLE APP_DATA_STR; DROP TABLE APP_DATA_LOB; DROP TABLE

Database	Create Tables	Drop Tables
	MC_FOLDER_ITEM (ID); ALTER TABLE MC_SNIPPET_REF_SNIPPET ADD CONSTRAINT FK_MC_SNIPPET_REF_SNIPPET_SNIPPET_ID FOREIGN KEY (SNIPPET_ID) REFERENCES MC_FOLDER_ITEM (ID); ALTER TABLE MC_SNIPPET_REF_TYPE ADD CONSTRAINT FK_MC_SNIPPET_REF_TYPE_SNIPPET_ID FOREIGN KEY (SNIPPET_ID) REFERENCES MC_FOLDER_ITEM (ID); ALTER TABLE MC_SNIPPET_TYPED_VAR ADD CONSTRAINT FK_MC_SNIPPET_TYPED_VAR_SNIPPET_ID FOREIGN KEY (SNIPPET_ID) REFERENCES MC_FOLDER_ITEM (ID); ALTER TABLE MC_SNIPPET_REF_VAR ADD CONSTRAINT FK_MC_SNIPPET_REF_VAR_SNIPPET_ID FOREIGN KEY (SNIPPET_ID) REFERENCES MC_FOLDER_ITEM (ID); ALTER TABLE MC_OAUTH_APPLICATION ADD CONSTRAINT FK_MC_OAUTH_APPLICATION_P_ID FOREIGN KEY (P_ID) REFERENCES MC_PROJECT (ID); ALTER TABLE MC_OAUTH_USER ADD CONSTRAINT FK_MC_OAUTH_USER_APPLICATION_ID FOREIGN KEY (APPLICATION_ID) REFERENCES MC_OAUTH_APPLICATION (ID); ALTER TABLE MC_OAUTH_USER ADD CONSTRAINT FK_MC_OAUTH_USER_USER_ID FOREIGN KEY (USER_ID) REFERENCES MC_USER (USER_NAME); ALTER TABLE MC_USER_GROUP_REL ADD CONSTRAINT FK_MC_USER_GROUP_REL_GROUP_ID FOREIGN KEY (GROUP_ID) REFERENCES MC_GROUP (ID); ALTER TABLE MC_USER_GROUP_REL ADD CONSTRAINT FK_MC_USER_GROUP_REL_USER_ID FOREIGN KEY (USER_ID) REFERENCES MC_USER (USER_NAME); ALTER TABLE MC_USER_PROP ADD CONSTRAINT FK_MC_USER_PROP_USER_ID FOREIGN KEY (USER_ID) REFERENCES MC_USER (USER_NAME); ALTER TABLE MC_TYPE_ATTR ADD CONSTRAINT FK_MC_TYPE_ATTR_TYPE_ID FOREIGN KEY (TYPE_ID) REFERENCES MC_FOLDER_ITEM (ID); ALTER TABLE MC_DATABASE ADD CONSTRAINT FK_MC_DATABASE_CLUSTER_SETTINGS_ID FOREIGN KEY (CLUSTER_SETTINGS_ID) REFERENCES MC_RS_CLUSTER_SETTINGS (ID); ALTER TABLE MC_PROXY_SERVER ADD CONSTRAINT FK_MC_PROXY_SERVER_CLUSTER_SETTINGS_ID FOREIGN KEY (CLUSTER_SETTINGS_ID) REFERENCES MC_RS_CLUSTER_SETTINGS (ID); ALTER TABLE APP_KAPPLET ADD CONSTRAINT FK_APP_KAPPLET_ICON_ID FOREIGN KEY (ICON_ID) REFERENCES APP_ICON (ID); ALTER TABLE APP_KAPPLET ADD CONSTRAINT FK_APP_KAPPLET_P_ID FOREIGN KEY (P_ID) REFERENCES MC_PROJECT (ID); ALTER TABLE APP_MASTER ADD CONSTRAINT FK_APP_MASTER_ID FOREIGN KEY (ID) REFERENCES APP_KAPPLET (ID); ALTER TABLE APP_INSTALLED ADD CONSTRAINT FK_APP_INSTALLED_USER_ID FOREIGN KEY (USER_ID) REFERENCES MC_USER (USER_NAME); ALTER TABLE APP_INSTALLED ADD CONSTRAINT FK_APP_INSTALLED_PARENT	APP_FLOW_ERROR; DROP TABLE APP_RUN; DROP TABLE APP_RUN_VALUE; DROP TABLE MC_FOLDER; DROP TABLE MC_ROOT_FOLDER; DROP TABLE MC_SUBFOLDER; DROP TABLE APP_RUN_APP_SCHEDULE; DROP TABLE APP_SEQUENCES;

Database	Create Tables	Drop Tables
	FOREIGN KEY (PARENT) REFERENCES APP_KAPPLET (ID); ALTER TABLE APP_INSTALLED ADD CONSTRAINT FK_APP_INSTALLED_ID FOREIGN KEY (ID) REFERENCES APP_KAPPLET (ID); ALTER TABLE MC_KAPPLET_PROP ADD CONSTRAINT FK_MC_KAPPLET_PROP_KAPPLET_ID FOREIGN KEY (KAPPLET_ID) REFERENCES APP_KAPPLET (ID); ALTER TABLE APP_SCHEDULE_ERROR ADD CONSTRAINT FK_APP_SCHEDULE_ERROR_SCHEDULE_ID FOREIGN KEY (SCHEDULE_ID) REFERENCES APP_SCHEDULE (ID); ALTER TABLE APP_SCHEDULE ADD CONSTRAINT FK_APP_SCHEDULE_APP_ID FOREIGN KEY (APP_ID) REFERENCES APP_KAPPLET (ID); ALTER TABLE APP_SCHEDULE_VALUE ADD CONSTRAINT FK_APP_SCHEDULE_VALUE_SCHEDULE_ID FOREIGN KEY (SCHEDULE_ID) REFERENCES APP_SCHEDULE (ID); ALTER TABLE APP_ACTION_EXECUTION ADD CONSTRAINT FK_APP_ACTION_EXECUTION_KAPPLET_RUN_ID FOREIGN KEY (KAPPLET_RUN_ID) REFERENCES APP_RUN (ID); ALTER TABLE APP_DATA_ROW ADD CONSTRAINT FK_APP_DATA_ROW_TABLE_ID FOREIGN KEY (TABLE_ID) REFERENCES APP_DATA_TABLE (ID); ALTER TABLE APP_DATA_TABLE ADD CONSTRAINT FK_APP_DATA_TABLE_FLOW_EXECUTION_ID FOREIGN KEY (FLOW_EXECUTION_ID) REFERENCES APP_ACTION_EXECUTION (ID); ALTER TABLE APP_DATA_STR ADD CONSTRAINT FK_APP_DATA_STR_RESULT_ID FOREIGN KEY (RESULT_ID) REFERENCES APP_DATA_ROW (ID); ALTER TABLE APP_DATA_LOB ADD CONSTRAINT FK_APP_DATA_LOB_RESULT_ID FOREIGN KEY (RESULT_ID) REFERENCES APP_DATA_ROW (ID); ALTER TABLE APP_FLOW_ERROR ADD CONSTRAINT FK_APP_FLOW_ERROR_EXECUTION_ID FOREIGN KEY (EXECUTION_ID) REFERENCES APP_ACTION_EXECUTION (ID); ALTER TABLE APP_RUN ADD CONSTRAINT FK_APP_RUN_KAPPLET_ID FOREIGN KEY (KAPPLET_ID) REFERENCES APP_KAPPLET (ID); ALTER TABLE APP_RUN_VALUE ADD CONSTRAINT FK_APP_RUN_VALUE_RUN_ID FOREIGN KEY (RUN_ID) REFERENCES APP_RUN (ID); ALTER TABLE MC_ROOT_FOLDER ADD CONSTRAINT FK_MC_ROOT_FOLDER_P_ID FOREIGN KEY (P_ID) REFERENCES MC_PROJECT (ID); ALTER TABLE MC_ROOT_FOLDER ADD CONSTRAINT FK_MC_ROOT_FOLDER_ID FOREIGN KEY (ID) REFERENCES MC_FOLDER (ID); ALTER TABLE MC_SUBFOLDER ADD CONSTRAINT FK_MC_SUBFOLDER_ID FOREIGN KEY (ID) REFERENCES MC_FOLDER (ID); ALTER TABLE MC_SUBFOLDER ADD CONSTRAINT FK_MC_SUBFOLDER_PARENT_ID FOREIGN KEY (PARENT_ID) REFERENCES MC_FOLDER (ID); ALTER TABLE APP_RUN_APP_SCHEDULE ADD CONSTRAINT FK_APP_RUN_APP_SCHEDULE_kappletRuns_ID FOREIGN KEY (kappletRuns_ID)	

Database	Create Tables	Drop Tables
	REFERENCES APP_RUN (ID); ALTER TABLE APP_RUN_APP_SCHEDULE ADD CONSTRAINT FK_APP_RUN_APP_SCHEDULE_schedules_ID FOREIGN KEY (schedules_ID) REFERENCES APP_SCHEDULE (ID); CREATE TABLE APP_SEQUENCES (TABLE_NAME VARCHAR(50) NOT NULL, NEXT_ID DECIMAL(38), PRIMARY KEY (TABLE_NAME)); INSERT INTO APP_SEQUENCES(TABLE_NAME, NEXT_ID) values ('DATA_STR', 0); INSERT INTO APP_SEQUENCES(TABLE_NAME, NEXT_ID) values ('DATA_LOB', 0); INSERT INTO APP_SEQUENCES(TABLE_NAME, NEXT_ID) values ('FLOW_ACTION_EXECUTION', 0); INSERT INTO APP_SEQUENCES(TABLE_NAME, NEXT_ID) values ('DATA_TABLE', 0); INSERT INTO APP_SEQUENCES(TABLE_NAME, NEXT_ID) values ('DATA_ROW', 0);	
Oracle	CREATE TABLE MC_SCHEDULE (ID NUMBER(19) NOT NULL, BAD_INPUT NUMBER(1) default 0 NULL, ACTIVE NUMBER(1) default 0 NULL, CREATEDBY NVARCHAR2(255) NULL, DIRTY NUMBER(1) default 0 NULL, EMAILS NVARCHAR2(255) NULL, MAXOBJECTSEXTRACTED NUMBER(10) NULL, MAXRUNTIME NUMBER(10) NULL, MODIFIEDBY NVARCHAR2(255) NULL, NAME NVARCHAR2(255) NOT NULL, NEXTRUN TIMESTAMP NULL, PREVIOUSRUN TIMESTAMP NULL, TOTALRUNS NUMBER(10) NULL, USEEMAILNOTIFICATION NUMBER(1) default 0 NULL, CLUSTER_ID NUMBER(19) NULL, P_ID NUMBER(19) NOT NULL, BLOCKJOB_ID NUMBER(19) NULL, PRIMARY KEY (ID)); CREATE TABLE MC_BLOCK (ID NUMBER(19) NOT NULL, DTYPE NVARCHAR2(31) NULL, NAME NVARCHAR2(255) NOT NULL, NAMEDBLOCK NUMBER(1) default 0 NULL, P_ID NUMBER(19) NOT NULL, PRIMARY KEY (ID)); CREATE TABLE MC_BLOCK_REFERENCE (ID NUMBER(19) NOT NULL, ENABLED NUMBER(1) default 0 NOT NULL, ORDER_VAL NUMBER(10) NOT NULL, NAME NVARCHAR2(255) NOT NULL, CHILD NUMBER(19) NULL, PARENT NUMBER(19) NOT NULL, PRIMARY KEY (ID)); CREATE TABLE MC_BLOCK_JOB (ID NUMBER(19) NOT NULL, ACTIVE NUMBER(1) default 0 NOT NULL, BLOCKINPUT NCLOB NULL, DISPLAY_NAMES NCLOB NOT NULL, BLOCK_ID NUMBER(19) NOT NULL, PRIMARY KEY (ID)); CREATE TABLE MC_ROBOT_BLOCK (ID NUMBER(19) NOT NULL, ROBOT_ID NUMBER(19) NULL, PRIMARY KEY (ID)); CREATE TABLE MC_MULTIPLE_ROBOT_BLOCK (ID NUMBER(19) NOT NULL, DISPLAY_NAME NVARCHAR2(255) NULL, STRATEGY NVARCHAR2(255) NULL, VALUE NVARCHAR2(255) NULL, PRIMARY KEY (ID)); CREATE TABLE MC_CONTAINER_BLOCK (ID NUMBER(19) NOT NULL, MAPPING BLOB NULL, PRIMARY KEY (ID)); CREATE TABLE MC_SEQUENTIAL_BLOCK (ID NUMBER(19) NOT NULL, PRIMARY KEY (ID)); CREATE TABLE	ALTER TABLE MC_SCHEDULE DROP CONSTRAINT FK_MC_SCHEDULE_BLOCKJOB_ID; ALTER TABLE MC_SCHEDULE DROP CONSTRAINT FK_MC_SCHEDULE_CLUSTER_ID; ALTER TABLE MC_SCHEDULE DROP CONSTRAINT FK_MC_SCHEDULE_P_ID; ALTER TABLE MC_SCHEDULE DROP CONSTRAINT UNQ_MC_SCHEDULE_0; ALTER TABLE MC_BLOCK DROP CONSTRAINT FK_MC_BLOCK_P_ID; ALTER TABLE MC_BLOCK DROP CONSTRAINT UNQ_MC_BLOCK_0; ALTER TABLE MC_BLOCK_REFERENCE DROP CONSTRAINT FK_MC_BLOCK_REFERENCE_PARENT; ALTER TABLE MC_BLOCK_REFERENCE DROP CONSTRAINT FK_MC_BLOCK_REFERENCE_CHILD; ALTER TABLE MC_BLOCK_REFERENCE DROP CONSTRAINT UNQ_MC_BLOCK_REFERENCE_0; ALTER TABLE MC_BLOCK_JOB DROP CONSTRAINT FK_MC_BLOCK_JOB_BLOCK_ID;

Database	Create Tables	Drop Tables
	MC_CONCURRENT_BLOCK (ID NUMBER(19) NOT NULL, PRIMARY KEY (ID)); CREATE TABLE MC_SCRIPT_BLOCK (ID NUMBER(19) NOT NULL, SCRIPT NVARCHAR2(255) NULL, PRIMARY KEY (ID)); CREATE TABLE MC_FOLDER_ITEM (ID NUMBER(19) NOT NULL, DTYPE NVARCHAR2(31) NULL, FOLDER NUMBER(19) NOT NULL, PRIMARY KEY (ID)); CREATE TABLE MC_ROBOT (ID NUMBER(19) NOT NULL, BYTESIZE NUMBER(10) NOT NULL, CONTAINSRETURNOBJECT NUMBER(1) default 0 NULL, CREATEDBY NVARCHAR2(255) NULL, LASTMODIFIED TIMESTAMP NOT NULL, MODIFIEDBY NVARCHAR2(255) NULL, NAME NVARCHAR2(255) NOT NULL, ROBOTBYTES BLOB NOT NULL, SHA_HASH NVARCHAR2(255) NOT NULL, VERSION NVARCHAR2(255) NOT NULL, FOLDER_EX NUMBER(19) NOT NULL, P_ID NUMBER(19) NOT NULL, PRIMARY KEY (ID)); CREATE TABLE MC_TYPE (ID NUMBER(19) NOT NULL, CREATEDBY NVARCHAR2(255) NULL, LASTMODIFIED TIMESTAMP NULL, MODIFIEDBY NVARCHAR2(255) NULL, NAME NVARCHAR2(255) NOT NULL, SHA_HASH NVARCHAR2(255) NOT NULL, TYPEBYTES BLOB NOT NULL, P_ID NUMBER(19) NOT NULL, PRIMARY KEY (ID)); CREATE TABLE MC_SNIPPET (ID NUMBER(19) NOT NULL, CREATEDBY NVARCHAR2(255) NULL, LASTMODIFIED TIMESTAMP NOT NULL, MODIFIEDBY NVARCHAR2(255) NULL, NAME NVARCHAR2(255) NOT NULL, SHA_HASH NVARCHAR2(255) NOT NULL, SNIPPETBYTES BLOB NOT NULL, VERSION NVARCHAR2(255) NOT NULL, P_ID NUMBER(19) NOT NULL, PRIMARY KEY (ID)); CREATE TABLE MC_ROBOSERVER (ID NUMBER(19) NOT NULL, HOST NVARCHAR2(255) NOT NULL, PORT NUMBER(10) NOT NULL, CLUSTER_ID NUMBER(19) NOT NULL, PRIMARY KEY (ID)); CREATE TABLE MC_SETTINGS (NAME NVARCHAR2(255) NOT NULL, VALUE NVARCHAR2(255) NULL, PRIMARY KEY (NAME)); CREATE TABLE MC_ROBOT_PARAMETER (ID NUMBER(19) NOT NULL, TYPENAME NVARCHAR2(255) NOT NULL, VARIABLENAME NVARCHAR2(255) NOT NULL, ROBOT_ID NUMBER(19) NOT NULL, PRIMARY KEY (ID)); CREATE TABLE MC_ROBOT_REF_TYPE (ID NUMBER(19) NOT NULL, ISINPUT NUMBER(1) default 0 NOT NULL, ISRETURNED NUMBER(1) default 0 NOT NULL, ISSTORED NUMBER(1) default 0 NOT NULL, NAME NVARCHAR2(255) NOT NULL, ROBOT_ID NUMBER(19) NOT NULL, PRIMARY KEY (ID)); CREATE TABLE MC_ROBOT_REF_SNIPPET (ID NUMBER(19) NOT NULL, NAME NVARCHAR2(255) NOT NULL, ROBOT_ID NUMBER(19) NOT NULL, PRIMARY KEY (ID)); CREATE TABLE MC_ROBOT_TYPED_VARIABLE (ID NUMBER(19) NOT NULL, TYPENAME NVARCHAR2(255) NOT NULL, VARIABLENAME NVARCHAR2(255) NOT NULL, ROBOT_ID NUMBER(19) NOT NULL, PRIMARY KEY (ID)); CREATE TABLE MC_ROBOT_REF_VAR (ID NUMBER(19) NOT NULL, ISRETURNED NUMBER(1)	ALTER TABLE MC_ROBOT_BLOCK DROP CONSTRAINT FK_MC_ROBOT_BLOCK_ID; ALTER TABLE MC_ROBOT_BLOCK DROP CONSTRAINT FK_MC_ROBOT_BLOCK_ROBOT_ID; ALTER TABLE MC_MULTIPLE_ROBOT_BLOCK DROP CONSTRAINT FK_MC_MULTIPLE_ROBOT_BLOCK_ID; ALTER TABLE MC_CONTAINER_BLOCK DROP CONSTRAINT FK_MC_CONTAINER_BLOCK_ID; ALTER TABLE MC_SEQUENTIAL_BLOCK DROP CONSTRAINT FK_MC_SEQUENTIAL_BLOCK_ID; ALTER TABLE MC_CONCURRENT_BLOCK DROP CONSTRAINT FK_MC_CONCURRENT_BLOCK_ID; ALTER TABLE MC_SCRIPT_BLOCK DROP CONSTRAINT FK_MC_SCRIPT_BLOCK_ID; ALTER TABLE MC_FOLDER_ITEM DROP CONSTRAINT FK_MC_FOLDER_ITEM_FOLDER; ALTER TABLE MC_ROBOT DROP CONSTRAINT FK_MC_ROBOT_FOLDER_EX; ALTER TABLE MC_ROBOT DROP CONSTRAINT FK_MC_ROBOT_ID; ALTER TABLE MC_ROBOT DROP CONSTRAINT FK_MC_ROBOT_P_ID; ALTER TABLE MC_ROBOT DROP CONSTRAINT UNQ_MC_ROBOT_0; ALTER TABLE MC_TYPE DROP CONSTRAINT FK_MC_TYPE_P_ID; ALTER TABLE MC_TYPE DROP CONSTRAINT FK_MC_TYPE_ID; ALTER TABLE MC_TYPE DROP

Database	Create Tables	Drop Tables
	default 0 NOT NULL, ISSTORED NUMBER(1) default 0 NOT NULL, NAME NVARCHAR2(255) NULL, ROBOT_ID NUMBER(19) NOT NULL, PRIMARY KEY (ID)); CREATE TABLE MC_RESOURCE_FILE (ID NUMBER(19) NOT NULL, BYTESIZE NUMBER(10) NOT NULL, BYTES BLOB NOT NULL, CREATEDBY NVARCHAR2(255) NULL, LASTMODIFIED TIMESTAMP NOT NULL, MODIFIEDBY NVARCHAR2(255) NULL, NAME NVARCHAR2(255) NOT NULL, SHA_HASH NVARCHAR2(255) NOT NULL, FOLDER_EX NUMBER(19) NOT NULL, P_ID NUMBER(19) NOT NULL, PRIMARY KEY (ID)); CREATE TABLE MC_PROJECT (ID NUMBER(19) NOT NULL, ACAO NVARCHAR2(255) NULL, AUTHENTICATE NUMBER(1) default 0 NULL, DESCRIPTION NVARCHAR2(255) NULL, NAME NVARCHAR2(255) NOT NULL UNIQUE, CLUSTER_ID NUMBER(19) NULL, PRIMARY KEY (ID)); CREATE TABLE MC_ROBOSERVER_CLUSTER (ID NUMBER(19) NOT NULL, KCU NUMBER(10) NULL, NAME NVARCHAR2(255) NOT NULL UNIQUE, type NUMBER(1) default 0 NOT NULL, USESSL NUMBER(1) default 0 NOT NULL, SETTINGS_ID NUMBER(19) NULL, PRIMARY KEY (ID)); CREATE TABLE MC_GROUP_TO_ROLE_MAP (ID NUMBER(19) NOT NULL, GROUP_NAME NVARCHAR2(200) NOT NULL, ROLE_NAME NVARCHAR2(40) NOT NULL, P_ID NUMBER(19) NOT NULL, PRIMARY KEY (ID)); CREATE TABLE MC_SNIPPET_PARAMETER (ID NUMBER(19) NOT NULL, TYPENAME NVARCHAR2(255) NOT NULL, VARIABLENAME NVARCHAR2(255) NOT NULL, SNIPPET_ID NUMBER(19) NOT NULL, PRIMARY KEY (ID)); CREATE TABLE MC_SNIPPET_REF_SNIPPET (ID NUMBER(19) NOT NULL, NAME NVARCHAR2(255) NOT NULL, SNIPPET_ID NUMBER(19) NOT NULL, PRIMARY KEY (ID)); CREATE TABLE MC_SNIPPET_REF_TYPE (ID NUMBER(19) NOT NULL, ISINPUT NUMBER(1) default 0 NOT NULL, ISRETURNED NUMBER(1) default 0 NOT NULL, ISSTORED NUMBER(1) default 0 NOT NULL, NAME NVARCHAR2(255) NULL, SNIPPET_ID NUMBER(19) NOT NULL, PRIMARY KEY (ID)); CREATE TABLE MC_SNIPPET_TYPED_VAR (ID NUMBER(19) NOT NULL, TYPENAME NVARCHAR2(255) NOT NULL, VARIABLENAME NVARCHAR2(255) NOT NULL, SNIPPET_ID NUMBER(19) NOT NULL, PRIMARY KEY (ID)); CREATE TABLE MC_SNIPPET_REF_VAR (ID NUMBER(19) NOT NULL, ISRETURNED NUMBER(1) default 0 NOT NULL, ISSTORED NUMBER(1) default 0 NOT NULL, NAME NVARCHAR2(255) NULL, SNIPPET_ID NUMBER(19) NOT NULL, PRIMARY KEY (ID)); CREATE TABLE MC_OAUTH_APPLICATION (ID NUMBER(19) NOT NULL, CALLBACKURL NVARCHAR2(255) NULL, CONSUMERKEY NVARCHAR2(255) NULL, CONSUMERSECRET NCLOB NULL, NAME NVARCHAR2(255) NOT NULL, SCOPE NVARCHAR2(255) NULL,	CONSTRAINT UNQ_MC_TYPE_0; ALTER TABLE MC_SNIPPET DROP CONSTRAINT FK_MC_SNIPPET_P_ID; ALTER TABLE MC_SNIPPET DROP CONSTRAINT FK_MC_SNIPPET_ID; ALTER TABLE MC_SNIPPET DROP CONSTRAINT UNQ_MC_SNIPPET_0; ALTER TABLE MC_ROBOSERVER DROP CONSTRAINT FK_MC_ROBOSERVER_CLUSTER_ID; ALTER TABLE MC_ROBOT_PARAMETER DROP CONSTRAINT FK_MC_ROBOT_PARAMETER_ROBOT_ID; ALTER TABLE MC_ROBOT_REF_TYPE DROP CONSTRAINT FK_MC_ROBOT_REF_TYPE_ROBOT_ID; ALTER TABLE MC_ROBOT_REF_SNIPPET DROP CONSTRAINT MC_ROBOT_REF_SNIPPET_ROBOT_ID; ALTER TABLE MC_ROBOT_TYPED_VARIABLE DROP CONSTRAINT MCROBOT_TYPED_VARIABLEROBOT_ID; ALTER TABLE MC_ROBOT_REF_VAR DROP CONSTRAINT FK_MC_ROBOT_REF_VAR_ROBOT_ID; ALTER TABLE MC_RESOURCE_FILE DROP CONSTRAINT FK_MC_RESOURCE_FILE_FOLDER_EX; ALTER TABLE MC_RESOURCE_FILE DROP CONSTRAINT FK_MC_RESOURCE_FILE_ID; ALTER TABLE MC_RESOURCE_FILE DROP CONSTRAINT FK_MC_RESOURCE_FILE_P_ID; ALTER TABLE MC_RESOURCE_FILE DROP CONSTRAINT UNQ_MC_RESOURCE_FILE_0; ALTER TABLE MC_PROJECT DROP CONSTRAINT FK_MC_PROJECT_CLUSTER_ID;

Database	Create Tables	Drop Tables
	SERVICEPROVIDER NVARCHAR2(255) NULL, P_ID NUMBER(19) NOT NULL, PRIMARY KEY (ID)); CREATE TABLE MC_OAUTH_USER (ID NUMBER(19) NOT NULL, ACCESSTOKEN NVARCHAR2(255) NULL, ACCESSTOKENSECRET NCLOB NULL, NAME NVARCHAR2(255) NOT NULL, REFRESHTOKEN NVARCHAR2(255) NULL, APPLICATION_ID NUMBER(19) NOT NULL, USER_ID NVARCHAR2(255) NULL, PRIMARY KEY (ID)); CREATE TABLE MC_DSLICENSE (TOKEN NVARCHAR2(255) NOT NULL, EXPIRATION NUMBER(19) NULL, IP NVARCHAR2(255) NULL, USERNAME NVARCHAR2(255) NULL, PRIMARY KEY (TOKEN)); CREATE TABLE MC_USER (USER_NAME NVARCHAR2(255) NOT NULL, ADMIN_USER NUMBER(1) default 0 NULL, CREATION_DATE TIMESTAMP NULL, EMAIL_ADDR NVARCHAR2(255) NULL, FULL_NAME NVARCHAR2(255) NULL, LAST_IP_ADDR NVARCHAR2(255) NULL, LOGIN_TIME TIMESTAMP NULL, LOGIN_COUNT NUMBER(10) NOT NULL, PASS_WORD NVARCHAR2(255) NULL, PRIMARY KEY (USER_NAME)); CREATE TABLE MC_GROUP (ID NUMBER(19) NOT NULL, DESCRIPTION NVARCHAR2(255) NULL, GROUP_NAME NVARCHAR2(255) NOT NULL, PRIMARY KEY (ID)); CREATE TABLE MC_USER_GROUP_REL (ID NUMBER(19) NOT NULL, GROUP_ID NUMBER(19) NOT NULL, USER_ID NVARCHAR2(255) NOT NULL, PRIMARY KEY (ID)); CREATE TABLE MC_USER_PROP (ID NUMBER(19) NOT NULL, IS_NULL NUMBER(1) default 0 NULL, PROP_NAME NVARCHAR2(255) NOT NULL, PROP_VALUE NCLOB NULL, USER_ID NVARCHAR2(255) NOT NULL, PRIMARY KEY (ID)); CREATE TABLE MC_TYPE_ATTR (ID NUMBER(19) NOT NULL, ATTR_TYPE NVARCHAR2(255) NULL, COMMENTS NCLOB NULL, DEFAULT_VALUE NCLOB NULL, NAME NVARCHAR2(255) NULL, REQUIRED_ATTR NUMBER(1) default 0 NULL, TYPE_ORDER NUMBER(10) NULL, TYPE_ID NUMBER(19) NOT NULL, PRIMARY KEY (ID)); CREATE TABLE MC_RS_CLUSTER_SETTINGS (ID NUMBER(19) NOT NULL, CLUSTER_SETTINGS_ID NVARCHAR2(255) NOT NULL, EMAIL_LOG_FROM_ADDRESS NVARCHAR2(255) NULL, LOG_INPUT_TO_DB NUMBER(1) default 0 NULL, LOG_INPUT_TO_LOG4J NUMBER(1) default 0 NULL, LOG_TO_DB NUMBER(1) default 0 NOT NULL, LOG_TO_EMAIL NUMBER(1) default 0 NOT NULL, LOG_TO_LOG4J NUMBER(1) default 0 NOT NULL, MAX_CONCURRENT_ROBOTS NUMBER(10) NULL, MAX_QUEUED_ROBOTS NUMBER(10) NULL, PROF_ENABLED NUMBER(1) default 0 NULL, PROF_FILE_OUTPUT_APPEND NUMBER(1) default 0 NULL, PROF_FILE_OUTPUT_FILE NVARCHAR2(255) NULL, PROF_LOG_PAGE_URL NUMBER(1) default 0 NULL, PROF_OUTPUT_TARGET NVARCHAR2(20) NULL, PROF_TYPE NVARCHAR2(20) NULL, PROF_THRESHOLD NUMBER(10) NULL,	ALTER TABLE MC_ROBOSERVER_CLUSTER DROP CONSTRAINT MCROBOSERVERCLUSTERSETTINGS_ID; ALTER TABLE MC_GROUP_TO_ROLE_MAP DROP CONSTRAINT FK_MC_GROUP_TO_ROLE_MAP_P_ID; ALTER TABLE MC_GROUP_TO_ROLE_MAP DROP CONSTRAINT UNQ_MC_GROUP_TO_ROLE_MAP_0; ALTER TABLE MC_SNIPPET_PARAMETER DROP CONSTRAINT MC_SNIPPET_PARAMETERSNIPPET_ID; ALTER TABLE MC_SNIPPET_REF_SNIPPET DROP CONSTRAINT MCSNIPPETREF_SNIPPETSNIPPET_ID; ALTER TABLE MC_SNIPPET_REF_TYPE DROP CONSTRAINT MC_SNIPPET_REF_TYPE_SNIPPET_ID; ALTER TABLE MC_SNIPPET_TYPED_VAR DROP CONSTRAINT MC_SNIPPET_TYPED_VARSNIPPET_ID; ALTER TABLE MC_SNIPPET_REF_VAR DROP CONSTRAINT MC_SNIPPET_REF_VAR_SNIPPET_ID; ALTER TABLE MC_OAUTH_APPLICATION DROP CONSTRAINT FK_MC_OAUTH_APPLICATION_P_ID; ALTER TABLE MC_OAUTH_APPLICATION DROP CONSTRAINT UNQ_MC_OAUTH_APPLICATION_0; ALTER TABLE MC_OAUTH_USER DROP CONSTRAINT MC_OAUTH_USER_APPLICATION_ID; ALTER TABLE MC_OAUTH_USER DROP CONSTRAINT FK_MC_OAUTH_USER_USER_ID; ALTER TABLE MC_OAUTH_USER DROP CONSTRAINT UNQ_MC_OAUTH_USER_0; ALTER TABLE MC_GROUP DROP CONSTRAINT UNQ_MC_GROUP_0; ALTER TABLE MC_USER_GROUP_REL

Database	Create Tables	Drop Tables
	EMAIL_LOG_TO_ADDRESS NVARCHAR2(255) NULL, PRIMARY KEY (ID)); CREATE TABLE MC_DATABASE (ID NUMBER(19) NOT NULL, HOST NVARCHAR2(255) NOT NULL, MAX_ACTIVE_CONNECTIONS NUMBER(10) NULL, MAX_IDLE_CONNECTIONS NUMBER(10) NULL, NAME NVARCHAR2(255) NOT NULL, PASSWORD NVARCHAR2(255) NULL, SCHEMA_NAME NVARCHAR2(255) NOT NULL, TYPE NVARCHAR2(255) NOT NULL, USERNAME NVARCHAR2(255) NULL, CLUSTER_SETTINGS_ID NUMBER(19) NOT NULL, PRIMARY KEY (ID)); CREATE TABLE MC_PROXY_SERVER (ID NUMBER(19) NOT NULL, EXCLUDED_HOST_NAMES NVARCHAR2(1000) NULL, HOST_NAME NVARCHAR2(255) NOT NULL, PASSWORD NVARCHAR2(255) NULL, PORT_NUMBER NUMBER(10) NULL, USERNAME NVARCHAR2(255) NULL, CLUSTER_SETTINGS_ID NUMBER(19) NOT NULL, PRIMARY KEY (ID)); CREATE TABLE MC_DATABASE_TYPE (ID NUMBER(19) NOT NULL, DRIVER_CLASS NVARCHAR2(255) NOT NULL, NAME NVARCHAR2(255) NOT NULL, URL_TEMPLATE NVARCHAR2(255) NOT NULL, VALIDATION_QUERY NVARCHAR2(255) NOT NULL, PRIMARY KEY (ID)); CREATE TABLE MC_JAR_FILE (ID NUMBER(19) NOT NULL, BYTES BLOB NULL, CREATED TIMESTAMP NULL, NAME NVARCHAR2(255) NOT NULL, FILE_SIZE NUMBER(10) NULL, PRIMARY KEY (ID)); CREATE TABLE APP_KAPPLET (ID NVARCHAR2(40) NOT NULL, DTYPE NVARCHAR2(31) NULL, CREATED_DATE TIMESTAMP NOT NULL, DESCRIPTION NCLOB NULL, NAME NVARCHAR2(255) NOT NULL, OPTIONS NCLOB NULL, P_ID NUMBER(19) NOT NULL, ICON_ID NVARCHAR2(255) NULL, PRIMARY KEY (ID)); CREATE TABLE APP_MASTER (ID NVARCHAR2(40) NOT NULL, CREATEDBY NVARCHAR2(255) NULL, ENABLED NUMBER(1) default 0 NOT NULL, FLOW NCLOB NULL, INITIAL_LABEL NUMBER(10) NULL, KAPPLET_TYPE NVARCHAR2(255) NOT NULL, LAST_INSTALLED TIMESTAMP NULL, LAST_MODIFIED TIMESTAMP NOT NULL, LAST_RUN TIMESTAMP NULL, MODIFIEDBY NVARCHAR2(255) NULL, PRIMARY KEY (ID)); CREATE TABLE APP_INSTALLED (ID NVARCHAR2(40) NOT NULL, FAVORITE NUMBER(1) default 0 NULL, LAST_RUN TIMESTAMP NULL, UPDATED_AT TIMESTAMP NULL, PARENT NVARCHAR2(40) NOT NULL, USER_ID NVARCHAR2(255) NOT NULL, PRIMARY KEY (ID)); CREATE TABLE APP_ICON (ID NVARCHAR2(255) NOT NULL, ICON BLOB NULL, filename NVARCHAR2(255) NULL, mimeType NVARCHAR2(255) NULL, PRIMARY KEY (ID)); CREATE TABLE APP_UPLOAD (ID NVARCHAR2(255) NOT NULL, UPLOAD BLOB NOT NULL, filename NVARCHAR2(255) NOT NULL, mimeType NVARCHAR2(255) NOT NULL, timestamp TIMESTAMP NOT NULL, username NVARCHAR2(255) NOT NULL, PRIMARY KEY (ID)); CREATE TABLE	DROP CONSTRAINT FK_MC_USER_GROUP_REL_GROUP_ID; ALTER TABLE MC_USER_GROUP_REL DROP CONSTRAINT FK_MC_USER_GROUP_REL_USER_ID; ALTER TABLE MC_USER_GROUP_REL DROP CONSTRAINT UNQ_MC_USER_GROUP_REL_0; ALTER TABLE MC_USER_PROP DROP CONSTRAINT FK_MC_USER_PROP_USER_ID; ALTER TABLE MC_USER_PROP DROP CONSTRAINT UNQ_MC_USER_PROP_0; ALTER TABLE MC_TYPE_ATTR DROP CONSTRAINT FK_MC_TYPE_ATTR_TYPE_ID; ALTER TABLE MC_DATABASE DROP CONSTRAINT MC_DATABASECLUSTER_SETTINGS_ID; ALTER TABLE MC_PROXY_SERVER DROP CONSTRAINT MCPROXYSERVERCLUSTERSETTINGSID; ALTER TABLE APP_KAPPLET DROP CONSTRAINT FK_APP_KAPPLET_ICON_ID; ALTER TABLE APP_KAPPLET DROP CONSTRAINT FK_APP_KAPPLET_P_ID; ALTER TABLE APP_MASTER DROP CONSTRAINT FK_APP_MASTER_ID; ALTER TABLE APP_INSTALLED DROP CONSTRAINT FK_APP_INSTALLED_USER_ID; ALTER TABLE APP_INSTALLED DROP CONSTRAINT FK_APP_INSTALLED_PARENT; ALTER TABLE APP_INSTALLED DROP CONSTRAINT FK_APP_INSTALLED_ID; ALTER TABLE MC_KAPPLET_PROP DROP CONSTRAINT FK_MC_KAPPLET_PROP_KAPPLET_ID;

Database	Create Tables	Drop Tables
	MC_KAPPLET_PROP (ID NUMBER(19) NOT NULL, IS_NULL NUMBER(1) default 0 NULL, PROP_NAME NVARCHAR2(255) NOT NULL, PROP_VALUE NCLOB NULL, KAPPLET_ID NVARCHAR2(40) NOT NULL, PRIMARY KEY (ID)); CREATE TABLE APP_SCHEDULE_ERROR (ID NUMBER(19) NOT NULL, ERROR NCLOB NULL, SCHEDULE_ID NUMBER(19) NOT NULL, PRIMARY KEY (ID)); CREATE TABLE APP_SCHEDULE (ID NUMBER(19) NOT NULL, DIRTY NUMBER(1) default 0 NULL, LASTRUN TIMESTAMP NULL, NEXTRUN TIMESTAMP NULL, PARAMETERS NCLOB NULL, APP_ID NVARCHAR2(40) NOT NULL, PRIMARY KEY (ID)); CREATE TABLE APP_SCHEDULE_VALUE (ID NVARCHAR2(255) NOT NULL, VALUE BLOB NOT NULL, SCHEDULE_ID NUMBER(19) NOT NULL, PRIMARY KEY (ID)); CREATE TABLE MC_GLOBAL_KAPPLET_PROP (PROP_NAME NVARCHAR2(255) NOT NULL, IS_NULL NUMBER(1) default 0 NULL, PROP_VALUE NCLOB NULL, PRIMARY KEY (PROP_NAME)); CREATE TABLE APP_ACTION_EXECUTION (ID NUMBER(19) NOT NULL, END_TIME TIMESTAMP NULL, LABEL NUMBER(10) NOT NULL, SEQ_ID NUMBER(10) NULL, START_TIME TIMESTAMP NULL, STATUS NVARCHAR2(255) NULL, TASK_IDS NVARCHAR2(255) NULL, VERSIONLOCK NUMBER(19) NULL, KAPPLET_RUN_ID NUMBER(19) NOT NULL, PRIMARY KEY (ID)); CREATE TABLE APP_DATA_ROW (ID NUMBER(19) NOT NULL, TABLE_ID NUMBER(19) NOT NULL, PRIMARY KEY (ID)); CREATE TABLE APP_DATA_TABLE (ID NUMBER(19) NOT NULL, TABLE_NAME NVARCHAR2(255) NOT NULL, FLOW_EXECUTION_ID NUMBER(19) NOT NULL, PRIMARY KEY (ID)); CREATE TABLE APP_DATA_STR (ID NUMBER(19) NOT NULL, NAME NVARCHAR2(255) NULL, VALUE NVARCHAR2(2000) NOT NULL, RESULT_ID NUMBER(19) NOT NULL, PRIMARY KEY (ID)); CREATE TABLE APP_DATA_LOB (ID NUMBER(19) NOT NULL, LARGE NUMBER(1) default 0 NULL, NAME NVARCHAR2(255) NULL, VALUE NCLOB NULL, RESULT_ID NUMBER(19) NOT NULL, PRIMARY KEY (ID)); CREATE TABLE APP_FLOW_ERROR (ID NUMBER(19) NOT NULL, ERROR NCLOB NULL, EXECUTION_ID NUMBER(19) NOT NULL, PRIMARY KEY (ID)); CREATE TABLE APP_RUN (ID NUMBER(19) NOT NULL, ABORTED NUMBER(1) default 0 NULL, FLOW NCLOB NULL, INITIAL_LABEL NUMBER(10) NOT NULL, LASTEXECUTIONPATHCHANGE NUMBER(19) NULL, OPTIONS NCLOB NULL, PARAMETERS NCLOB NULL, SIGNATURES NCLOB NULL, VERSIONLOCK NUMBER(19) NULL, KAPPLET_ID NVARCHAR2(40) NOT NULL, PRIMARY KEY (ID)); CREATE TABLE APP_RUN_VALUE (ID NVARCHAR2(255) NOT NULL, VALUE BLOB NOT NULL, RUN_ID NUMBER(19) NOT NULL, PRIMARY KEY (ID)); CREATE TABLE MC_FOLDER (ID NUMBER(19) NOT NULL, DTYPE NVARCHAR2(31) NULL, PATH NVARCHAR2(255)	ALTER TABLE MC_KAPPLET_PROP DROP CONSTRAINT UNQ_MC_KAPPLET_PROP_0; ALTER TABLE APP_SCHEDULE_ERROR DROP CONSTRAINT APP_SCHEDULE_ERROR_SCHEDULE_ID; ALTER TABLE APP_SCHEDULE DROP CONSTRAINT FK_APP_SCHEDULE_APP_ID; ALTER TABLE APP_SCHEDULE_VALUE DROP CONSTRAINT APP_SCHEDULE_VALUE_SCHEDULE_ID; ALTER TABLE APP_ACTION_EXECUTION DROP CONSTRAINT APPACTIONEXECUTIONKAPPLETRUNID; ALTER TABLE APP_ACTION_EXECUTION DROP CONSTRAINT UNQ_APP_ACTION_EXECUTION_0; ALTER TABLE APP_DATA_ROW DROP CONSTRAINT FK_APP_DATA_ROW_TABLE_ID; ALTER TABLE APP_DATA_TABLE DROP CONSTRAINT APPDATA_TABLEFLOW_EXECUTION_ID; ALTER TABLE APP_DATA_TABLE DROP CONSTRAINT UNQ_APP_DATA_TABLE_0; ALTER TABLE APP_DATA_STR DROP CONSTRAINT FK_APP_DATA_STR_RESULT_ID; ALTER TABLE APP_DATA_LOB DROP CONSTRAINT FK_APP_DATA_LOB_RESULT_ID; ALTER TABLE APP_FLOW_ERROR DROP CONSTRAINT FK_APP_FLOW_ERROR_EXECUTION_ID; ALTER TABLE APP_RUN DROP CONSTRAINT FK_APP_RUN_KAPPLET_ID; ALTER TABLE APP_RUN_VALUE DROP CONSTRAINT FK_APP_RUN_VALUE_RUN_ID; ALTER TABLE MC_ROOT_FOLDER

Database	Create Tables	Drop Tables
	NULL, PRIMARY KEY (ID)); CREATE TABLE MC_ROOT_FOLDER (ID NUMBER(19) NOT NULL, P_ID NUMBER(19) NOT NULL, PRIMARY KEY (ID)); CREATE TABLE MC_SUBFOLDER (ID NUMBER(19) NOT NULL, NAME NVARCHAR2(255) NOT NULL, PARENT_ID NUMBER(19) NOT NULL, PRIMARY KEY (ID)); CREATE TABLE APP_RUN_APP_SCHEDULE (schedules_ID NUMBER(19) NOT NULL, kappletRuns_ID NUMBER(19) NOT NULL, PRIMARY KEY (schedules_ID, kappletRuns_ID)); ALTER TABLE MC_SCHEDULE ADD CONSTRAINT UNQ_MC_SCHEDULE_0 UNIQUE (P_ID, NAME); ALTER TABLE MC_BLOCK ADD CONSTRAINT UNQ_MC_BLOCK_0 UNIQUE (P_ID, NAME); ALTER TABLE MC_BLOCK_REFERENCE ADD CONSTRAINT UNQ_MC_BLOCK_REFERENCE_0 UNIQUE (PARENT, NAME); ALTER TABLE MC_ROBOT ADD CONSTRAINT UNQ_MC_ROBOT_0 UNIQUE (P_ID, NAME, FOLDER_EX); ALTER TABLE MC_TYPE ADD CONSTRAINT UNQ_MC_TYPE_0 UNIQUE (P_ID, NAME); ALTER TABLE MC_SNIPPET ADD CONSTRAINT UNQ_MC_SNIPPET_0 UNIQUE (P_ID, NAME); ALTER TABLE MC_RESOURCE_FILE ADD CONSTRAINT UNQ_MC_RESOURCE_FILE_0 UNIQUE (P_ID, NAME, FOLDER_EX); ALTER TABLE MC_GROUP_TO_ROLE_MAP ADD CONSTRAINT UNQ_MC_GROUP_TO_ROLE_MAP_0 UNIQUE (P_ID, ROLE_NAME, GROUP_NAME); ALTER TABLE MC_OAUTH_APPLICATION ADD CONSTRAINT UNQ_MC_OAUTH_APPLICATION_0 UNIQUE (P_ID, NAME); ALTER TABLE MC_OAUTH_USER ADD CONSTRAINT UNQ_MC_OAUTH_USER_0 UNIQUE (APPLICATION_ID, NAME); ALTER TABLE MC_GROUP ADD CONSTRAINT UNQ_MC_GROUP_0 UNIQUE (GROUP_NAME); ALTER TABLE MC_USER_GROUP_REL ADD CONSTRAINT UNQ_MC_USER_GROUP_REL_0 UNIQUE (USER_ID, GROUP_ID); ALTER TABLE MC_USER_PROP ADD CONSTRAINT UNQ_MC_USER_PROP_0 UNIQUE (ID, PROP_NAME); ALTER TABLE MC_KAPPLET_PROP ADD CONSTRAINT UNQ_MC_KAPPLET_PROP_0 UNIQUE (ID, PROP_NAME); ALTER TABLE APP_ACTION_EXECUTION ADD CONSTRAINT UNQ_APP_ACTION_EXECUTION_0 UNIQUE (KAPPLET_RUN_ID, LABEL); ALTER TABLE APP_DATA_TABLE ADD CONSTRAINT UNQ_APP_DATA_TABLE_0 UNIQUE (FLOW_EXECUTION_ID, TABLE_NAME); ALTER TABLE MC_SUBFOLDER ADD CONSTRAINT UNQ_MC_SUBFOLDER_0 UNIQUE (NAME, PARENT_ID); ALTER TABLE MC_SCHEDULE ADD CONSTRAINT FK_MC_SCHEDULE_BLOCKJOB_ID FOREIGN KEY (BLOCKJOB_ID) REFERENCES MC_BLOCK_JOB (ID); ALTER TABLE MC_SCHEDULE ADD CONSTRAINT FK_MC_SCHEDULE_CLUSTER_ID FOREIGN KEY (CLUSTER_ID) REFERENCES MC_ROBOSERVER_CLUSTER (ID); ALTER TABLE MC_SCHEDULE ADD CONSTRAINT FK_MC_SCHEDULE_P_ID FOREIGN KEY (P_ID) REFERENCES MC_PROJECT (ID); ALTER TABLE	DROP CONSTRAINT FK_MC_ROOT_FOLDER_P_ID; ALTER TABLE MC_ROOT_FOLDER DROP CONSTRAINT FK_MC_ROOT_FOLDER_ID; ALTER TABLE MC_SUBFOLDER DROP CONSTRAINT FK_MC_SUBFOLDER_ID; ALTER TABLE MC_SUBFOLDER DROP CONSTRAINT FK_MC_SUBFOLDER_PARENT_ID; ALTER TABLE MC_SUBFOLDER DROP CONSTRAINT UNQ_MC_SUBFOLDER_0; ALTER TABLE APP_RUN_APP_SCHEDULE DROP CONSTRAINT APPRUNAPPSCHEDULEkappletRunsID; ALTER TABLE APP_RUN_APP_SCHEDULE DROP CONSTRAINT APPRUNAPP_SCHEDULEschedules_ID; DROP TABLE MC_SCHEDULE CASCADE CONSTRAINTS; DROP TABLE MC_BLOCK CASCADE CONSTRAINTS; DROP TABLE MC_BLOCK_REFERENCE CASCADE CONSTRAINTS; DROP TABLE MC_BLOCK_JOB CASCADE CONSTRAINTS; DROP TABLE MC_ROBOT_BLOCK CASCADE CONSTRAINTS; DROP TABLE MC_MULTIPLE_ROBOT_BLOCK CASCADE CONSTRAINTS; DROP TABLE MC_CONTAINER_BLOCK CASCADE CONSTRAINTS; DROP TABLE MC_SEQUENTIAL_BLOCK CASCADE CONSTRAINTS; DROP TABLE

Database	Create Tables	Drop Tables
	MC_BLOCK ADD CONSTRAINT FK_MC_BLOCK_P_ID FOREIGN KEY (P_ID) REFERENCES MC_PROJECT (ID); ALTER TABLE MC_BLOCK_REFERENCE ADD CONSTRAINT FK_MC_BLOCK_REFERENCE_PARENT FOREIGN KEY (PARENT) REFERENCES MC_BLOCK (ID); ALTER TABLE MC_BLOCK_REFERENCE ADD CONSTRAINT FK_MC_BLOCK_REFERENCE_CHILD FOREIGN KEY (CHILD) REFERENCES MC_BLOCK (ID); ALTER TABLE MC_BLOCK_JOB ADD CONSTRAINT FK_MC_BLOCK_JOB_BLOCK_ID FOREIGN KEY (BLOCK_ID) REFERENCES MC_BLOCK (ID); ALTER TABLE MC_ROBOT_BLOCK ADD CONSTRAINT FK_MC_ROBOT_BLOCK_ID FOREIGN KEY (ID) REFERENCES MC_BLOCK (ID); ALTER TABLE MC_ROBOT_BLOCK ADD CONSTRAINT FK_MC_ROBOT_BLOCK_ROBOT_ID FOREIGN KEY (ROBOT_ID) REFERENCES MC_FOLDER_ITEM (ID); ALTER TABLE MC_MULTIPLE_ROBOT_BLOCK ADD CONSTRAINT FK_MC_MULTIPLE_ROBOT_BLOCK_ID FOREIGN KEY (ID) REFERENCES MC_BLOCK (ID); ALTER TABLE MC_CONTAINER_BLOCK ADD CONSTRAINT FK_MC_CONTAINER_BLOCK_ID FOREIGN KEY (ID) REFERENCES MC_BLOCK (ID); ALTER TABLE MC_SEQUENTIAL_BLOCK ADD CONSTRAINT FK_MC_SEQUENTIAL_BLOCK_ID FOREIGN KEY (ID) REFERENCES MC_BLOCK (ID); ALTER TABLE MC_CONCURRENT_BLOCK ADD CONSTRAINT FK_MC_CONCURRENT_BLOCK_ID FOREIGN KEY (ID) REFERENCES MC_BLOCK (ID); ALTER TABLE MC_SCRIPT_BLOCK ADD CONSTRAINT FK_MC_SCRIPT_BLOCK_ID FOREIGN KEY (ID) REFERENCES MC_BLOCK (ID); ALTER TABLE MC_FOLDER_ITEM ADD CONSTRAINT FK_MC_FOLDER_ITEM_FOLDER FOREIGN KEY (FOLDER) REFERENCES MC_FOLDER (ID); ALTER TABLE MC_ROBOT ADD CONSTRAINT FK_MC_ROBOT_FOLDER_EX FOREIGN KEY (FOLDER_EX) REFERENCES MC_FOLDER (ID); ALTER TABLE MC_ROBOT ADD CONSTRAINT FK_MC_ROBOT_ID FOREIGN KEY (ID) REFERENCES MC_FOLDER_ITEM (ID); ALTER TABLE MC_ROBOT ADD CONSTRAINT FK_MC_ROBOT_P_ID FOREIGN KEY (P_ID) REFERENCES MC_PROJECT (ID); ALTER TABLE MC_TYPE ADD CONSTRAINT FK_MC_TYPE_P_ID FOREIGN KEY (P_ID) REFERENCES MC_PROJECT (ID); ALTER TABLE MC_TYPE ADD CONSTRAINT FK_MC_TYPE_ID FOREIGN KEY (ID) REFERENCES MC_FOLDER_ITEM (ID); ALTER TABLE MC_SNIPPET ADD CONSTRAINT FK_MC_SNIPPET_P_ID FOREIGN KEY (P_ID) REFERENCES MC_PROJECT (ID); ALTER TABLE MC_SNIPPET ADD CONSTRAINT FK_MC_SNIPPET_ID FOREIGN KEY (ID) REFERENCES MC_FOLDER_ITEM (ID); ALTER TABLE MC_ROBOSERVER ADD CONSTRAINT FK_MC_ROBOSERVER_CLUSTER_ID FOREIGN KEY (CLUSTER_ID) REFERENCES MC_ROBOSERVER_CLUSTER (ID); ALTER TABLE MC_ROBOT_PARAMETER ADD CONSTRAINT	MC_CONCURRENT_BLOCK CASCADE CONSTRAINTS; DROP TABLE MC_SCRIPT_BLOCK CASCADE CONSTRAINTS; DROP TABLE MC_FOLDER_ITEM CASCADE CONSTRAINTS; DROP TABLE MC_ROBOT CASCADE CONSTRAINTS; DROP TABLE MC_TYPE CASCADE CONSTRAINTS; DROP TABLE MC_SNIPPET CASCADE CONSTRAINTS; DROP TABLE MC_ROBOSERVER CASCADE CONSTRAINTS; DROP TABLE MC_SETTINGS CASCADE CONSTRAINTS; DROP TABLE MC_ROBOT_PARAMETER CASCADE CONSTRAINTS; DROP TABLE MC_ROBOT_REF_TYPE CASCADE CONSTRAINTS; DROP TABLE MC_ROBOT_REF_SNIPPET CASCADE CONSTRAINTS; DROP TABLE MC_ROBOT_TYPED_VARIABLE CASCADE CONSTRAINTS; DROP TABLE MC_ROBOT_REF_VAR CASCADE CONSTRAINTS; DROP TABLE MC_RESOURCE_FILE CASCADE CONSTRAINTS; DROP TABLE MC_PROJECT CASCADE CONSTRAINTS; DROP TABLE MC_ROBOSERVER_CLUSTER CASCADE CONSTRAINTS;

Database	Create Tables	Drop Tables
	FK_MC_ROBOT_PARAMETER_ROBOT_ID FOREIGN KEY (ROBOT_ID) REFERENCES MC_FOLDER_ITEM (ID); ALTER TABLE MC_ROBOT_REF_TYPE ADD CONSTRAINT FK_MC_ROBOT_REF_TYPE_ROBOT_ID FOREIGN KEY (ROBOT_ID) REFERENCES MC_FOLDER_ITEM (ID); ALTER TABLE MC_ROBOT_REF_SNIPPET ADD CONSTRAINT MC_ROBOT_REF_SNIPPET_ROBOT_ID FOREIGN KEY (ROBOT_ID) REFERENCES MC_FOLDER_ITEM (ID); ALTER TABLE MC_ROBOT_TYPED_VARIABLE ADD CONSTRAINT MCROBOT_TYPED_VARIABLEROBOT_ID FOREIGN KEY (ROBOT_ID) REFERENCES MC_FOLDER_ITEM (ID); ALTER TABLE MC_ROBOT_REF_VAR ADD CONSTRAINT FK_MC_ROBOT_REF_VAR_ROBOT_ID FOREIGN KEY (ROBOT_ID) REFERENCES MC_FOLDER_ITEM (ID); ALTER TABLE MC_RESOURCE_FILE ADD CONSTRAINT FK_MC_RESOURCE_FILE_FOLDER_EX FOREIGN KEY (FOLDER_EX) REFERENCES MC_FOLDER (ID); ALTER TABLE MC_RESOURCE_FILE ADD CONSTRAINT FK_MC_RESOURCE_FILE_ID FOREIGN KEY (ID) REFERENCES MC_FOLDER_ITEM (ID); ALTER TABLE MC_RESOURCE_FILE ADD CONSTRAINT FK_MC_RESOURCE_FILE_P_ID FOREIGN KEY (P_ID) REFERENCES MC_PROJECT (ID); ALTER TABLE MC_PROJECT ADD CONSTRAINT FK_MC_PROJECT_CLUSTER_ID FOREIGN KEY (CLUSTER_ID) REFERENCES MC_ROBOSERVER_CLUSTER (ID); ALTER TABLE MC_ROBOSERVER_CLUSTER ADD CONSTRAINT MCROBOSERVERCLUSTERSETTINGS_ID FOREIGN KEY (SETTINGS_ID) REFERENCES MC_RS_CLUSTER_SETTINGS (ID); ALTER TABLE MC_GROUP_TO_ROLE_MAP ADD CONSTRAINT FK_MC_GROUP_TO_ROLE_MAP_P_ID FOREIGN KEY (P_ID) REFERENCES MC_PROJECT (ID); ALTER TABLE MC_SNIPPET_PARAMETER ADD CONSTRAINT MC_SNIPPET_PARAMETERSNIPPET_ID FOREIGN KEY (SNIPPET_ID) REFERENCES MC_FOLDER_ITEM (ID); ALTER TABLE MC_SNIPPET_REF_SNIPPET ADD CONSTRAINT MCSNIPPETREF_SNIPPETSNIPPET_ID FOREIGN KEY (SNIPPET_ID) REFERENCES MC_FOLDER_ITEM (ID); ALTER TABLE MC_SNIPPET_REF_TYPE ADD CONSTRAINT MC_SNIPPET_REF_TYPE_SNIPPET_ID FOREIGN KEY (SNIPPET_ID) REFERENCES MC_FOLDER_ITEM (ID); ALTER TABLE MC_SNIPPET_TYPED_VAR ADD CONSTRAINT MC_SNIPPET_TYPED_VARSNIPPET_ID FOREIGN KEY (SNIPPET_ID) REFERENCES MC_FOLDER_ITEM (ID); ALTER TABLE MC_SNIPPET_REF_VAR ADD CONSTRAINT MC_SNIPPET_REF_VAR_SNIPPET_ID FOREIGN KEY (SNIPPET_ID) REFERENCES MC_FOLDER_ITEM (ID); ALTER TABLE MC_OAUTH_APPLICATION ADD CONSTRAINT FK_MC_OAUTH_APPLICATION_P_ID FOREIGN KEY (P_ID) REFERENCES MC_PROJECT (ID); ALTER TABLE MC_OAUTH_USER ADD CONSTRAINT MC_OAUTH_USER_APPLICATION_ID FOREIGN KEY (APPLICATION_ID) REFERENCES	DROP TABLE MC_GROUP_TO_ROLE_MAP CASCADE CONSTRAINTS; DROP TABLE MC_SNIPPET_PARAMETER CASCADE CONSTRAINTS; DROP TABLE MC_SNIPPET_REF_SNIPPET CASCADE CONSTRAINTS; DROP TABLE MC_SNIPPET_REF_TYPE CASCADE CONSTRAINTS; DROP TABLE MC_SNIPPET_TYPED_VAR CASCADE CONSTRAINTS; DROP TABLE MC_SNIPPET_REF_VAR CASCADE CONSTRAINTS; DROP TABLE MC_OAUTH_APPLICATION CASCADE CONSTRAINTS; DROP TABLE MC_OAUTH_USER CASCADE CONSTRAINTS; DROP TABLE MC_DSLICENSE CASCADE CONSTRAINTS; DROP TABLE MC_USER CASCADE CONSTRAINTS; DROP TABLE MC_GROUP CASCADE CONSTRAINTS; DROP TABLE MC_USER_GROUP_REL CASCADE CONSTRAINTS; DROP TABLE MC_USER_PROP CASCADE CONSTRAINTS; DROP TABLE MC_TYPE_ATTR CASCADE CONSTRAINTS; DROP TABLE MC_RS_CLUSTER_SETTINGS CASCADE CONSTRAINTS; DROP

Database	Create Tables	Drop Tables
	MC_OAUTH_APPLICATION (ID); ALTER TABLE MC_OAUTH_USER ADD CONSTRAINT FK_MC_OAUTH_USER_USER_ID FOREIGN KEY (USER_ID) REFERENCES MC_USER (USER_NAME); ALTER TABLE MC_USER_GROUP_REL ADD CONSTRAINT FK_MC_USER_GROUP_REL_GROUP_ID FOREIGN KEY (GROUP_ID) REFERENCES MC_GROUP (ID); ALTER TABLE MC_USER_GROUP_REL ADD CONSTRAINT FK_MC_USER_GROUP_REL_USER_ID FOREIGN KEY (USER_ID) REFERENCES MC_USER (USER_NAME); ALTER TABLE MC_USER_PROP ADD CONSTRAINT FK_MC_USER_PROP_USER_ID FOREIGN KEY (USER_ID) REFERENCES MC_USER (USER_NAME); ALTER TABLE MC_TYPE_ATTR ADD CONSTRAINT FK_MC_TYPE_ATTR_TYPE_ID FOREIGN KEY (TYPE_ID) REFERENCES MC_FOLDER_ITEM (ID); ALTER TABLE MC_DATABASE ADD CONSTRAINT MC_DATABASECLUSTER_SETTINGS_ID FOREIGN KEY (CLUSTER_SETTINGS_ID) REFERENCES MC_RS_CLUSTER_SETTINGS (ID); ALTER TABLE MC_PROXY_SERVER ADD CONSTRAINT MCPROXYSERVERCLUSTERSETTINGSID FOREIGN KEY (CLUSTER_SETTINGS_ID) REFERENCES MC_RS_CLUSTER_SETTINGS (ID); ALTER TABLE APP_KAPPLET ADD CONSTRAINT FK_APP_KAPPLET_ICON_ID FOREIGN KEY (ICON_ID) REFERENCES APP_ICON (ID); ALTER TABLE APP_KAPPLET ADD CONSTRAINT FK_APP_KAPPLET_P_ID FOREIGN KEY (P_ID) REFERENCES MC_PROJECT (ID); ALTER TABLE APP_MASTER ADD CONSTRAINT FK_APP_MASTER_ID FOREIGN KEY (ID) REFERENCES APP_KAPPLET (ID); ALTER TABLE APP_INSTALLED ADD CONSTRAINT FK_APP_INSTALLED_USER_ID FOREIGN KEY (USER_ID) REFERENCES MC_USER (USER_NAME); ALTER TABLE APP_INSTALLED ADD CONSTRAINT FK_APP_INSTALLED_PARENT FOREIGN KEY (PARENT) REFERENCES APP_KAPPLET (ID); ALTER TABLE APP_INSTALLED ADD CONSTRAINT FK_APP_INSTALLED_ID FOREIGN KEY (ID) REFERENCES APP_KAPPLET (ID); ALTER TABLE MC_KAPPLET_PROP ADD CONSTRAINT FK_MC_KAPPLET_PROP_KAPPLET_ID FOREIGN KEY (KAPPLET_ID) REFERENCES APP_KAPPLET (ID); ALTER TABLE APP_SCHEDULE_ERROR ADD CONSTRAINT APP_SCHEDULE_ERROR_SCHEDULE_ID FOREIGN KEY (SCHEDULE_ID) REFERENCES APP_SCHEDULE (ID); ALTER TABLE APP_SCHEDULE ADD CONSTRAINT FK_APP_SCHEDULE_APP_ID FOREIGN KEY (APP_ID) REFERENCES APP_KAPPLET (ID); ALTER TABLE APP_SCHEDULE_VALUE ADD CONSTRAINT APP_SCHEDULE_VALUE_SCHEDULE_ID FOREIGN KEY (SCHEDULE_ID) REFERENCES APP_SCHEDULE (ID); ALTER TABLE APP_ACTION_EXECUTION ADD CONSTRAINT APPACTIONEXECUTIONKAPPLETRUNID FOREIGN KEY (KAPPLET_RUN_ID) REFERENCES APP_RUN (ID); ALTER TABLE APP_DATA_ROW ADD CONSTRAINT	TABLE MC_DATABASE CASCADE CONSTRAINTS; DROP TABLE MC_PROXY_SERVER CASCADE CONSTRAINTS; DROP TABLE MC_DATABASE_TYPE CASCADE CONSTRAINTS; DROP TABLE MC_JAR_FILE CASCADE CONSTRAINTS; DROP TABLE APP_KAPPLET CASCADE CONSTRAINTS; DROP TABLE APP_MASTER CASCADE CONSTRAINTS; DROP TABLE APP_INSTALLED CASCADE CONSTRAINTS; DROP TABLE APP_ICON CASCADE CONSTRAINTS; DROP TABLE APP_UPLOAD CASCADE CONSTRAINTS; DROP TABLE MC_KAPPLET_PROP CASCADE CONSTRAINTS; DROP TABLE APP_SCHEDULE_ERROR CASCADE CONSTRAINTS; DROP TABLE APP_SCHEDULE CASCADE CONSTRAINTS; DROP TABLE APP_SCHEDULE_VALUE CASCADE CONSTRAINTS; DROP TABLE MC_GLOBAL_KAPPLET_PROP CASCADE CONSTRAINTS; DROP TABLE APP_ACTION_EXECUTION CASCADE CONSTRAINTS; DROP TABLE APP_DATA_ROW CASCADE CONSTRAINTS;

Database	Create Tables	Drop Tables
	FK_APP_DATA_ROW_TABLE_ID FOREIGN KEY (TABLE_ID) REFERENCES APP_DATA_TABLE (ID); ALTER TABLE APP_DATA_TABLE ADD CONSTRAINT APPDATA_TABLEFLOW_EXECUTION_ID FOREIGN KEY (FLOW_EXECUTION_ID) REFERENCES APP_ACTION_EXECUTION (ID); ALTER TABLE APP_DATA_STR ADD CONSTRAINT FK_APP_DATA_STR_RESULT_ID FOREIGN KEY (RESULT_ID) REFERENCES APP_DATA_ROW (ID); ALTER TABLE APP_DATA_LOB ADD CONSTRAINT FK_APP_DATA_LOB_RESULT_ID FOREIGN KEY (RESULT_ID) REFERENCES APP_DATA_ROW (ID); ALTER TABLE APP_FLOW_ERROR ADD CONSTRAINT FK_APP_FLOW_ERROR_EXECUTION_ID FOREIGN KEY (EXECUTION_ID) REFERENCES APP_ACTION_EXECUTION (ID); ALTER TABLE APP_RUN ADD CONSTRAINT FK_APP_RUN_KAPPLET_ID FOREIGN KEY (KAPPLET_ID) REFERENCES APP_KAPPLET (ID); ALTER TABLE APP_RUN_VALUE ADD CONSTRAINT FK_APP_RUN_VALUE_RUN_ID FOREIGN KEY (RUN_ID) REFERENCES APP_RUN (ID); ALTER TABLE MC_ROOT_FOLDER ADD CONSTRAINT FK_MC_ROOT_FOLDER_P_ID FOREIGN KEY (P_ID) REFERENCES MC_PROJECT (ID); ALTER TABLE MC_ROOT_FOLDER ADD CONSTRAINT FK_MC_ROOT_FOLDER_ID FOREIGN KEY (ID) REFERENCES MC_FOLDER (ID); ALTER TABLE MC_SUBFOLDER ADD CONSTRAINT FK_MC_SUBFOLDER_ID FOREIGN KEY (ID) REFERENCES MC_FOLDER (ID); ALTER TABLE MC_SUBFOLDER ADD CONSTRAINT FK_MC_SUBFOLDER_PARENT_ID FOREIGN KEY (PARENT_ID) REFERENCES MC_FOLDER (ID); ALTER TABLE APP_RUN_APP_SCHEDULE ADD CONSTRAINT APPRUNAPPSCHEDULEkappletRunsID FOREIGN KEY (kappletRuns_ID) REFERENCES APP_RUN (ID); ALTER TABLE APP_RUN_APP_SCHEDULE ADD CONSTRAINT APPRUNAPP_SCHEDULEschedules_ID FOREIGN KEY (schedules_ID) REFERENCES APP_SCHEDULE (ID); CREATE SEQUENCE SEQ_GEN_SEQUENCE INCREMENT BY 50 START WITH 50; CREATE TABLE APP_SEQUENCES (TABLE_NAME NVARCHAR2(50) NOT NULL, NEXT_ID NUMBER(38) NULL, PRIMARY KEY (TABLE_NAME)); INSERT INTO APP_SEQUENCES(TABLE_NAME, NEXT_ID) values ('DATA_LOB', 0); INSERT INTO APP_SEQUENCES(TABLE_NAME, NEXT_ID) values ('FLOW_ACTION_EXECUTION', 0); INSERT INTO APP_SEQUENCES(TABLE_NAME, NEXT_ID) values ('DATA_ROW', 0); INSERT INTO APP_SEQUENCES(TABLE_NAME, NEXT_ID) values ('DATA_STR', 0); INSERT INTO APP_SEQUENCES(TABLE_NAME, NEXT_ID) values ('DATA_TABLE', 0);	DROP TABLE APP_DATA_TABLE CASCADE CONSTRAINTS; DROP TABLE APP_DATA_STR CASCADE CONSTRAINTS; DROP TABLE APP_DATA_LOB CASCADE CONSTRAINTS; DROP TABLE APP_FLOW_ERROR CASCADE CONSTRAINTS; DROP TABLE APP_RUN CASCADE CONSTRAINTS; DROP TABLE APP_RUN_VALUE CASCADE CONSTRAINTS; DROP TABLE MC_FOLDER CASCADE CONSTRAINTS; DROP TABLE MC_ROOT_FOLDER CASCADE CONSTRAINTS; DROP TABLE MC_SUBFOLDER CASCADE CONSTRAINTS; DROP TABLE APP_RUN_APP_SCHEDULE CASCADE CONSTRAINTS; DROP SEQUENCE SEQ_GEN_SEQUENCE; DROP TABLE APP_SEQUENCES;
Microsoft SQL Server	CREATE TABLE MC_SCHEDULE (ID NUMERIC(19) NOT NULL, BAD_INPUT BIT default 0 NULL, ACTIVE BIT default 0 NULL, CREATEDBY NVARCHAR(255)	ALTER TABLE MC_SCHEDULE DROP CONSTRAINT

Database	Create Tables	Drop Tables
	NULL, DIRTY BIT default 0 NULL, EMAILS NVARCHAR(255) NULL, MAXOBJECTSEXTRACTED INTEGER NULL, MAXRUNTIME INTEGER NULL, MODIFIEDBY NVARCHAR(255) NULL, NAME NVARCHAR(255) NOT NULL, NEXTRUN DATETIME NULL, PREVIOUSRUN DATETIME NULL, TOTALRUNS INTEGER NULL, USEEMAILNOTIFICATION BIT default 0 NULL, CLUSTER_ID NUMERIC(19) NULL, P_ID NUMERIC(19) NOT NULL, BLOCKJOB_ID NUMERIC(19) NULL, PRIMARY KEY (ID)); CREATE TABLE MC_BLOCK (ID NUMERIC(19) IDENTITY NOT NULL, DTYPE NVARCHAR(31) NULL, NAME NVARCHAR(255) NOT NULL, NAMEDBLOCK BIT default 0 NULL, P_ID NUMERIC(19) NOT NULL, PRIMARY KEY (ID)); CREATE TABLE MC_BLOCK_REFERENCE (ID NUMERIC(19) IDENTITY NOT NULL, ENABLED BIT default 0 NOT NULL, ORDER_VAL INTEGER NOT NULL, NAME NVARCHAR(255) NOT NULL, CHILD NUMERIC(19) NULL, PARENT NUMERIC(19) NOT NULL, PRIMARY KEY (ID)); CREATE TABLE MC_BLOCK_JOB (ID NUMERIC(19) IDENTITY NOT NULL, ACTIVE BIT default 0 NOT NULL, BLOCKINPUT NTEXT NULL, DISPLAY_NAMES NTEXT NOT NULL, BLOCK_ID NUMERIC(19) NOT NULL, PRIMARY KEY (ID)); CREATE TABLE MC_ROBOT_BLOCK (ID NUMERIC(19) NOT NULL, ROBOT_ID NUMERIC(19) NULL, PRIMARY KEY (ID)); CREATE TABLE MC_MULTIPLE_ROBOT_BLOCK (ID NUMERIC(19) NOT NULL, DISPLAY_NAME NVARCHAR(255) NULL, STRATEGY NVARCHAR(255) NULL, VALUE NVARCHAR(255) NULL, PRIMARY KEY (ID)); CREATE TABLE MC_CONTAINER_BLOCK (ID NUMERIC(19) NOT NULL, MAPPING IMAGE NULL, PRIMARY KEY (ID)); CREATE TABLE MC_SEQUENTIAL_BLOCK (ID NUMERIC(19) NOT NULL, PRIMARY KEY (ID)); CREATE TABLE MC_CONCURRENT_BLOCK (ID NUMERIC(19) NOT NULL, PRIMARY KEY (ID)); CREATE TABLE MC_SCRIPT_BLOCK (ID NUMERIC(19) NOT NULL, SCRIPT NVARCHAR(255) NULL, PRIMARY KEY (ID)); CREATE TABLE MC_FOLDER_ITEM (ID NUMERIC(19) IDENTITY NOT NULL, DTYPE NVARCHAR(31) NULL, FOLDER NUMERIC(19) NOT NULL, PRIMARY KEY (ID)); CREATE TABLE MC_ROBOT (ID NUMERIC(19) NOT NULL, BYTESIZE INTEGER NOT NULL, CONTAINSRETURNOBJECT BIT default 0 NULL, CREATEDBY NVARCHAR(255) NULL, LASTMODIFIED DATETIME NOT NULL, MODIFIEDBY NVARCHAR(255) NULL, NAME NVARCHAR(255) NOT NULL, ROBOTBYTES IMAGE NOT NULL, SHA_HASH NVARCHAR(255) NOT NULL, VERSION NVARCHAR(255) NOT NULL, FOLDER_EX NUMERIC(19) NOT NULL, P_ID NUMERIC(19) NOT NULL, PRIMARY KEY (ID)); CREATE TABLE MC_TYPE (ID NUMERIC(19) NOT NULL, CREATEDBY NVARCHAR(255) NULL, LASTMODIFIED DATETIME NULL, MODIFIEDBY	MC_SCHEDULEBLOCKJOB_ID; ALTER TABLE MC_SCHEDULE DROP CONSTRAINT FK_MC_SCHEDULE_P_ID; ALTER TABLE MC_SCHEDULE DROP CONSTRAINT MC_SCHEDULE_CLUSTER_ID; ALTER TABLE MC_SCHEDULE DROP CONSTRAINT UNQ_MC_SCHEDULE_0; ALTER TABLE MC_BLOCK DROP CONSTRAINT FK_MC_BLOCK_P_ID; ALTER TABLE MC_BLOCK DROP CONSTRAINT UNQ_MC_BLOCK_0; ALTER TABLE MC_BLOCK_REFERENCE DROP CONSTRAINT MCBLOCK_REFERENCECHILD; ALTER TABLE MC_BLOCK_REFERENCE DROP CONSTRAINT MCBLOCKREFERENCEPARENT; ALTER TABLE MC_BLOCK_REFERENCE DROP CONSTRAINT MC_BLOCK_REFERENCE0; ALTER TABLE MC_BLOCK_JOB DROP CONSTRAINT MC_BLOCK_JOB_BLOCK_ID; ALTER TABLE MC_ROBOT_BLOCK DROP CONSTRAINT FK_MC_ROBOT_BLOCK_ID; ALTER TABLE MC_ROBOT_BLOCK DROP CONSTRAINT MC_ROBOT_BLOCKROBOT_ID; ALTER TABLE MC_MULTIPLE_ROBOT_BLOCK DROP CONSTRAINT MCMULTIPLEROBOTBLOCKID; ALTER TABLE MC_CONTAINER_BLOCK DROP CONSTRAINT MC_CONTAINER_BLOCK_ID; ALTER TABLE MC_SEQUENTIAL_BLOCK DROP CONSTRAINT MC_SEQUENTIAL_BLOCK_ID; ALTER TABLE

Database	Create Tables	Drop Tables
	NVARCHAR(255) NULL, NAME NVARCHAR(255) NOT NULL, SHA_HASH NVARCHAR(255) NOT NULL, TYPEBYTES IMAGE NOT NULL, P_ID NUMERIC(19) NOT NULL, PRIMARY KEY (ID)); CREATE TABLE MC_SNIPPET (ID NUMERIC(19) NOT NULL, CREATEDBY NVARCHAR(255) NULL, LASTMODIFIED DATETIME NOT NULL, MODIFIEDBY NVARCHAR(255) NULL, NAME NVARCHAR(255) NOT NULL, SHA_HASH NVARCHAR(255) NOT NULL, SNIPPETBYTES IMAGE NOT NULL, VERSION NVARCHAR(255) NOT NULL, P_ID NUMERIC(19) NOT NULL, PRIMARY KEY (ID)); CREATE TABLE MC_ROBOSERVER (ID NUMERIC(19) IDENTITY NOT NULL, HOST NVARCHAR(255) NOT NULL, PORT INTEGER NOT NULL, CLUSTER_ID NUMERIC(19) NOT NULL, PRIMARY KEY (ID)); CREATE TABLE MC_SETTINGS (NAME NVARCHAR(255) NOT NULL, VALUE NVARCHAR(255) NULL, PRIMARY KEY (NAME)); CREATE TABLE MC_ROBOT_PARAMETER (ID NUMERIC(19) IDENTITY NOT NULL, TYPENAME NVARCHAR(255) NOT NULL, VARIABLENAME NVARCHAR(255) NOT NULL, ROBOT_ID NUMERIC(19) NOT NULL, PRIMARY KEY (ID)); CREATE TABLE MC_ROBOT_REF_TYPE (ID NUMERIC(19) IDENTITY NOT NULL, ISINPUT BIT default 0 NOT NULL, ISRETURNED BIT default 0 NOT NULL, ISSTORED BIT default 0 NOT NULL, NAME NVARCHAR(255) NOT NULL, ROBOT_ID NUMERIC(19) NOT NULL, PRIMARY KEY (ID)); CREATE TABLE MC_ROBOT_REF_SNIPPET (ID NUMERIC(19) IDENTITY NOT NULL, NAME NVARCHAR(255) NOT NULL, ROBOT_ID NUMERIC(19) NOT NULL, PRIMARY KEY (ID)); CREATE TABLE MC_ROBOT_TYPED_VARIABLE (ID NUMERIC(19) IDENTITY NOT NULL, TYPENAME NVARCHAR(255) NOT NULL, VARIABLENAME NVARCHAR(255) NOT NULL, ROBOT_ID NUMERIC(19) NOT NULL, PRIMARY KEY (ID)); CREATE TABLE MC_ROBOT_REF_VAR (ID NUMERIC(19) IDENTITY NOT NULL, ISRETURNED BIT default 0 NOT NULL, ISSTORED BIT default 0 NOT NULL, NAME NVARCHAR(255) NULL, ROBOT_ID NUMERIC(19) NOT NULL, PRIMARY KEY (ID)); CREATE TABLE MC_RESOURCE_FILE (ID NUMERIC(19) NOT NULL, BYTESIZE INTEGER NOT NULL, BYTES IMAGE NOT NULL, CREATEDBY NVARCHAR(255) NULL, LASTMODIFIED DATETIME NOT NULL, MODIFIEDBY NVARCHAR(255) NULL, NAME NVARCHAR(255) NOT NULL, SHA_HASH NVARCHAR(255) NOT NULL, FOLDER_EX NUMERIC(19) NOT NULL, P_ID NUMERIC(19) NOT NULL, PRIMARY KEY (ID)); CREATE TABLE MC_PROJECT (ID NUMERIC(19) IDENTITY NOT NULL, ACAO NVARCHAR(255) NULL, AUTHENTICATE BIT default 0 NULL, DESCRIPTION NVARCHAR(255) NULL, NAME NVARCHAR(255) NOT NULL UNIQUE, CLUSTER_ID NUMERIC(19) NULL, PRIMARY KEY (ID)); CREATE TABLE MC_ROBOSERVER_CLUSTER (ID NUMERIC(19)	MC_CONCURRENT_BLOCK DROP CONSTRAINT MC_CONCURRENT_BLOCK_ID; ALTER TABLE MC_SCRIPT_BLOCK DROP CONSTRAINT FK_MC_SCRIPT_BLOCK_ID; ALTER TABLE MC_FOLDER_ITEM DROP CONSTRAINT MC_FOLDER_ITEM_FOLDER; ALTER TABLE MC_ROBOT DROP CONSTRAINT FK_MC_ROBOT_FOLDER_EX; ALTER TABLE MC_ROBOT DROP CONSTRAINT FK_MC_ROBOT_ID; ALTER TABLE MC_ROBOT DROP CONSTRAINT FK_MC_ROBOT_P_ID; ALTER TABLE MC_ROBOT DROP CONSTRAINT UNQ_MC_ROBOT_0; ALTER TABLE MC_TYPE DROP CONSTRAINT FK_MC_TYPE_P_ID; ALTER TABLE MC_TYPE DROP CONSTRAINT FK_MC_TYPE_ID; ALTER TABLE MC_TYPE DROP CONSTRAINT UNQ_MC_TYPE_0; ALTER TABLE MC_SNIPPET DROP CONSTRAINT FK_MC_SNIPPET_P_ID; ALTER TABLE MC_SNIPPET DROP CONSTRAINT FK_MC_SNIPPET_ID; ALTER TABLE MC_SNIPPET DROP CONSTRAINT UNQ_MC_SNIPPET_0; ALTER TABLE MC_ROBOSERVER DROP CONSTRAINT MCROBOSERVERCLUSTER_ID; ALTER TABLE MC_ROBOT_PARAMETER DROP CONSTRAINT

Database	Create Tables	Drop Tables
	IDENTITY NOT NULL, KCU INTEGER NULL, NAME NVARCHAR(255) NOT NULL UNIQUE, type BIT default 0 NOT NULL, USESSL BIT default 0 NOT NULL, SETTINGS_ID NUMERIC(19) NULL, PRIMARY KEY (ID)); CREATE TABLE MC_GROUP_TO_ROLE_MAP (ID NUMERIC(19) IDENTITY NOT NULL, GROUP_NAME NVARCHAR(200) NOT NULL, ROLE_NAME NVARCHAR(40) NOT NULL, P_ID NUMERIC(19) NOT NULL, PRIMARY KEY (ID)); CREATE TABLE MC_SNIPPET_PARAMETER (ID NUMERIC(19) IDENTITY NOT NULL, TYPENAME NVARCHAR(255) NOT NULL, VARIABLENAME NVARCHAR(255) NOT NULL, SNIPPET_ID NUMERIC(19) NOT NULL, PRIMARY KEY (ID)); CREATE TABLE MC_SNIPPET_REF_SNIPPET (ID NUMERIC(19) IDENTITY NOT NULL, NAME NVARCHAR(255) NOT NULL, SNIPPET_ID NUMERIC(19) NOT NULL, PRIMARY KEY (ID)); CREATE TABLE MC_SNIPPET_REF_TYPE (ID NUMERIC(19) IDENTITY NOT NULL, ISINPUT BIT default 0 NOT NULL, ISRETURNED BIT default 0 NOT NULL, ISSTORED BIT default 0 NOT NULL, NAME NVARCHAR(255) NULL, SNIPPET_ID NUMERIC(19) NOT NULL, PRIMARY KEY (ID)); CREATE TABLE MC_SNIPPET_TYPED_VAR (ID NUMERIC(19) IDENTITY NOT NULL, TYPENAME NVARCHAR(255) NOT NULL, VARIABLENAME NVARCHAR(255) NOT NULL, SNIPPET_ID NUMERIC(19) NOT NULL, PRIMARY KEY (ID)); CREATE TABLE MC_SNIPPET_REF_VAR (ID NUMERIC(19) IDENTITY NOT NULL, ISRETURNED BIT default 0 NOT NULL, ISSTORED BIT default 0 NOT NULL, NAME NVARCHAR(255) NULL, SNIPPET_ID NUMERIC(19) NOT NULL, PRIMARY KEY (ID)); CREATE TABLE MC_OAUTH_APPLICATION (ID NUMERIC(19) IDENTITY NOT NULL, CALLBACKURL NVARCHAR(255) NULL, CONSUMERKEY NVARCHAR(255) NULL, CONSUMERSECRET NTEXT NULL, NAME NVARCHAR(255) NOT NULL, SCOPE NVARCHAR(255) NULL, SERVICEPROVIDER NVARCHAR(255) NULL, P_ID NUMERIC(19) NOT NULL, PRIMARY KEY (ID)); CREATE TABLE MC_OAUTH_USER (ID NUMERIC(19) NOT NULL, ACCESSTOKEN NVARCHAR(255) NULL, ACCESSTOKENSECRET NTEXT NULL, NAME NVARCHAR(255) NOT NULL, REFRESHTOKEN NVARCHAR(255) NULL, APPLICATION_ID NUMERIC(19) NOT NULL, USER_ID NVARCHAR(255) NULL, PRIMARY KEY (ID)); CREATE TABLE MC_DSLICENSE (TOKEN NVARCHAR(255) NOT NULL, EXPIRATION NUMERIC(19) NULL, IP NVARCHAR(255) NULL, USERNAME NVARCHAR(255) NULL, PRIMARY KEY (TOKEN)); CREATE TABLE MC_USER (USER_NAME NVARCHAR(255) NOT NULL, ADMIN_USER BIT default 0 NULL, CREATION_DATE DATETIME NULL, EMAIL_ADDR NVARCHAR(255) NULL, FULL_NAME NVARCHAR(255) NULL, LAST_IP_ADDR NVARCHAR(255) NULL,	MCROBOTPARAMETERRBOTID; ALTER TABLE MC_ROBOT_REF_TYPE DROP CONSTRAINT MCROBOTREFTYPEROBOT_ID; ALTER TABLE MC_ROBOT_REF_SNIPPET DROP CONSTRAINT MCRBOTREFSNIPPETRBOTID; ALTER TABLE MC_ROBOT_TYPED_VARIABLE DROP CONSTRAINT MCRBTTYPEDVARIABLERBTD; ALTER TABLE MC_ROBOT_REF_VAR DROP CONSTRAINT MCROBOTREF_VARROBOT_ID; ALTER TABLE MC_RESOURCE_FILE DROP CONSTRAINT MCRESOURCEFILEFOLDEREX; ALTER TABLE MC_RESOURCE_FILE DROP CONSTRAINT MC_RESOURCE_FILE_P_ID; ALTER TABLE MC_RESOURCE_FILE DROP CONSTRAINT FK_MC_RESOURCE_FILE_ID; ALTER TABLE MC_RESOURCE_FILE DROP CONSTRAINT UNQ_MC_RESOURCE_FILE_0; ALTER TABLE MC_PROJECT DROP CONSTRAINT MC_PROJECT_CLUSTER_ID; ALTER TABLE MC_ROBOSERVER_CLUSTER DROP CONSTRAINT MCRBSRVRCLUSTERSTTNGSD; ALTER TABLE MC_GROUP_TO_ROLE_MAP DROP CONSTRAINT MCGROUPTO_ROLE_MAPP_ID; ALTER TABLE MC_GROUP_TO_ROLE_MAP DROP CONSTRAINT MC_GROUP_TO_ROLE_MAP0; ALTER TABLE MC_SNIPPET_PARAMETER DROP CONSTRAINT MCSNPPTPARAMETERSNPPTD; ALTER TABLE MC_SNIPPET_REF_SNIPPET DROP CONSTRAINT MCSNPPTRFSNIPPETSNPPTD; ALTER TABLE

Database	Create Tables	Drop Tables
	LOGIN_TIME DATETIME NULL, LOGIN_COUNT INTEGER NOT NULL, PASS_WORD NVARCHAR(255) NULL, PRIMARY KEY (USER_NAME)); CREATE TABLE MC_GROUP (ID NUMERIC(19) IDENTITY NOT NULL, DESCRIPTION NVARCHAR(255) NULL, GROUP_NAME NVARCHAR(255) NOT NULL, PRIMARY KEY (ID)); CREATE TABLE MC_USER_GROUP_REL (ID NUMERIC(19) IDENTITY NOT NULL, GROUP_ID NUMERIC(19) NOT NULL, USER_ID NVARCHAR(255) NOT NULL, PRIMARY KEY (ID)); CREATE TABLE MC_USER_PROP (ID NUMERIC(19) IDENTITY NOT NULL, IS_NULL BIT default 0 NULL, PROP_NAME NVARCHAR(255) NOT NULL, PROP_VALUE NTEXT NULL, USER_ID NVARCHAR(255) NOT NULL, PRIMARY KEY (ID)); CREATE TABLE MC_TYPE_ATTR (ID NUMERIC(19) IDENTITY NOT NULL, ATTR_TYPE NVARCHAR(255) NULL, COMMENTS NTEXT NULL, DEFAULT_VALUE NTEXT NULL, NAME NVARCHAR(255) NULL, REQUIRED_ATTR BIT default 0 NULL, TYPE_ORDER INTEGER NULL, TYPE_ID NUMERIC(19) NOT NULL, PRIMARY KEY (ID)); CREATE TABLE MC_RS_CLUSTER_SETTINGS (ID NUMERIC(19) IDENTITY NOT NULL, CLUSTER_SETTINGS_ID NVARCHAR(255) NOT NULL, EMAIL_LOG_FROM_ADDRESS NVARCHAR(255) NULL, LOG_INPUT_TO_DB BIT default 0 NULL, LOG_INPUT_TO_LOG4J BIT default 0 NULL, LOG_TO_DB BIT default 0 NOT NULL, LOG_TO_EMAIL BIT default 0 NOT NULL, LOG_TO_LOG4J BIT default 0 NOT NULL, MAX_CONCURRENT_ROBOTS INTEGER NULL, MAX_QUEUED_ROBOTS INTEGER NULL, PROF_ENABLED BIT default 0 NULL, PROF_FILE_OUTPUT_APPEND BIT default 0 NULL, PROF_FILE_OUTPUT_FILE NVARCHAR(255) NULL, PROF_LOG_PAGE_URL BIT default 0 NULL, PROF_OUTPUT_TARGET NVARCHAR(20) NULL, PROF_TYPE NVARCHAR(20) NULL, PROF_THRESHOLD INTEGER NULL, EMAIL_LOG_TO_ADDRESS NVARCHAR(255) NULL, PRIMARY KEY (ID)); CREATE TABLE MC_DATABASE (ID NUMERIC(19) IDENTITY NOT NULL, HOST NVARCHAR(255) NOT NULL, MAX_ACTIVE_CONNECTIONS INTEGER NULL, MAX_IDLE_CONNECTIONS INTEGER NULL, NAME NVARCHAR(255) NOT NULL, PASSWORD NVARCHAR(255) NULL, SCHEMA_NAME NVARCHAR(255) NOT NULL, TYPE NVARCHAR(255) NOT NULL, USERNAME NVARCHAR(255) NULL, CLUSTER_SETTINGS_ID NUMERIC(19) NOT NULL, PRIMARY KEY (ID)); CREATE TABLE MC_PROXY_SERVER (ID NUMERIC(19) IDENTITY NOT NULL, EXCLUDED_HOST_NAMES NVARCHAR(1000) NULL, HOST_NAME NVARCHAR(255) NOT NULL, PASSWORD NVARCHAR(255) NULL, PORT_NUMBER INTEGER NULL, USERNAME NVARCHAR(255) NULL, CLUSTER_SETTINGS_ID NUMERIC(19) NOT NULL, PRIMARY KEY (ID)); CREATE TABLE MC_DATABASE_TYPE (ID NUMERIC(19) IDENTITY	MC_SNIPPET_REF_TYPE DROP CONSTRAINT MCSNPPETREFTYPESNPPTID; ALTER TABLE MC_SNIPPET_TYPED_VAR DROP CONSTRAINT MCSNPPTTYPEDVARSNPPTID; ALTER TABLE MC_SNIPPET_REF_VAR DROP CONSTRAINT MCSNPPETREFVARSNPPETID; ALTER TABLE MC_OAUTH_APPLICATION DROP CONSTRAINT MCOAUTHAPPLICATIONP_ID; ALTER TABLE MC_OAUTH_APPLICATION DROP CONSTRAINT MC_OAUTH_APPLICATION0; ALTER TABLE MC_OAUTH_USER DROP CONSTRAINT MCAUTHUSERPPLICATIONID; ALTER TABLE MC_OAUTH_USER DROP CONSTRAINT MC_OAUTH_USER_USER_ID; ALTER TABLE MC_OAUTH_USER DROP CONSTRAINT UNQ_MC_OAUTH_USER_0; ALTER TABLE MC_GROUP DROP CONSTRAINT UNQ_MC_GROUP_0; ALTER TABLE MC_USER_GROUP_REL DROP CONSTRAINT MCUSERGROUP_RELUSER_ID; ALTER TABLE MC_USER_GROUP_REL DROP CONSTRAINT MCUSERGROUPRELGROUP_ID; ALTER TABLE MC_USER_GROUP_REL DROP CONSTRAINT MC_USER_GROUP_REL0; ALTER TABLE MC_USER_PROP DROP CONSTRAINT MC_USER_PROP_USER_ID; ALTER TABLE MC_USER_PROP DROP CONSTRAINT UNQ_MC_USER_PROP_0; ALTER TABLE MC_TYPE_ATTR DROP CONSTRAINT

Database	Create Tables	Drop Tables
	NOT NULL, DRIVER_CLASS NVARCHAR(255) NOT NULL, NAME NVARCHAR(255) NOT NULL, URL_TEMPLATE NVARCHAR(255) NOT NULL, VALIDATION_QUERY NVARCHAR(255) NOT NULL, PRIMARY KEY (ID)); CREATE TABLE MC_JAR_FILE (ID NUMERIC(19) IDENTITY NOT NULL, BYTES IMAGE NULL, CREATED DATETIME NULL, NAME NVARCHAR(255) NOT NULL, FILE_SIZE INTEGER NULL, PRIMARY KEY (ID)); CREATE TABLE APP_KAPPLET (ID NVARCHAR(40) NOT NULL, DTYPE NVARCHAR(31) NULL, CREATED_DATE DATETIME NOT NULL, DESCRIPTION NTEXT NULL, NAME NVARCHAR(255) NOT NULL, OPTIONS NTEXT NULL, P_ID NUMERIC(19) NOT NULL, ICON_ID NVARCHAR(255) NULL, PRIMARY KEY (ID)); CREATE TABLE APP_MASTER (ID NVARCHAR(40) NOT NULL, CREATEDBY NVARCHAR(255) NULL, ENABLED BIT default 0 NOT NULL, FLOW NTEXT NULL, INITIAL_LABEL INTEGER NULL, KAPPLET_TYPE NVARCHAR(255) NOT NULL, LAST_INSTALLED DATETIME NULL, LAST_MODIFIED DATETIME NOT NULL, LAST_RUN DATETIME NULL, MODIFIEDBY NVARCHAR(255) NULL, PRIMARY KEY (ID)); CREATE TABLE APP_INSTALLED (ID NVARCHAR(40) NOT NULL, FAVORITE BIT default 0 NULL, LAST_RUN DATETIME NULL, UPDATED_AT DATETIME NULL, PARENT NVARCHAR(40) NOT NULL, USER_ID NVARCHAR(255) NOT NULL, PRIMARY KEY (ID)); CREATE TABLE APP_ICON (ID NVARCHAR(255) NOT NULL, ICON IMAGE NULL, filename NVARCHAR(255) NULL, mimeType NVARCHAR(255) NULL, PRIMARY KEY (ID)); CREATE TABLE APP_UPLOAD (ID NVARCHAR(255) NOT NULL, UPLOAD IMAGE NOT NULL, filename NVARCHAR(255) NOT NULL, mimeType NVARCHAR(255) NOT NULL, timestamp DATETIME NOT NULL, username NVARCHAR(255) NOT NULL, PRIMARY KEY (ID)); CREATE TABLE MC_KAPPLET_PROP (ID NUMERIC(19) IDENTITY NOT NULL, IS_NULL BIT default 0 NULL, PROP_NAME NVARCHAR(255) NOT NULL, PROP_VALUE NTEXT NULL, KAPPLET_ID NVARCHAR(40) NOT NULL, PRIMARY KEY (ID)); CREATE TABLE APP_SCHEDULE_ERROR (ID NUMERIC(19) IDENTITY NOT NULL, ERROR NTEXT NULL, SCHEDULE_ID NUMERIC(19) NOT NULL, PRIMARY KEY (ID)); CREATE TABLE APP_SCHEDULE (ID NUMERIC(19) NOT NULL, DIRTY BIT default 0 NULL, LASTRUN DATETIME NULL, NEXTRUN DATETIME NULL, PARAMETERS NTEXT NULL, APP_ID NVARCHAR(40) NOT NULL, PRIMARY KEY (ID)); CREATE TABLE APP_SCHEDULE_VALUE (ID NVARCHAR(255) NOT NULL, VALUE IMAGE NOT NULL, SCHEDULE_ID NUMERIC(19) NOT NULL, PRIMARY KEY (ID)); CREATE TABLE MC_GLOBAL_KAPPLET_PROP (PROP_NAME NVARCHAR(255) NOT NULL, IS_NULL BIT default 0 NULL, PROP_VALUE NTEXT NULL, PRIMARY KEY (PROP_NAME)); CREATE TABLE	MC_TYPE_ATTR_TYPE_ID; ALTER TABLE MC_DATABASE DROP CONSTRAINT MCDTBSECLSTRSETTINGSID; ALTER TABLE MC_PROXY_SERVER DROP CONSTRAINT MCPRXYSRVRCLSTRSTTNGSD; ALTER TABLE APP_KAPPLET DROP CONSTRAINT FK_APP_KAPPLET_ICON_ID; ALTER TABLE APP_KAPPLET DROP CONSTRAINT FK_APP_KAPPLET_P_ID; ALTER TABLE APP_MASTER DROP CONSTRAINT FK_APP_MASTER_ID; ALTER TABLE APP_INSTALLED DROP CONSTRAINT APP_INSTALLED_USER_ID; ALTER TABLE APP_INSTALLED DROP CONSTRAINT APP_INSTALLED_PARENT; ALTER TABLE APP_INSTALLED DROP CONSTRAINT FK_APP_INSTALLED_ID; ALTER TABLE MC_KAPPLET_PROP DROP CONSTRAINT MCKAPPLETPROPKAPPLETID; ALTER TABLE MC_KAPPLET_PROP DROP CONSTRAINT UNQ_MC_KAPPLET_PROP_0; ALTER TABLE APP_SCHEDULE_ERROR DROP CONSTRAINT PPSCHDULEERRORSCHDLEID; ALTER TABLE APP_SCHEDULE DROP CONSTRAINT FK_APP_SCHEDULE_APP_ID; ALTER TABLE APP_SCHEDULE_VALUE DROP CONSTRAINT PPSCHDULEVALUESCHDLEID; ALTER TABLE APP_ACTION_EXECUTION DROP CONSTRAINT PPCTNXECUTIONKPPLTRNID; ALTER TABLE

Database	Create Tables	Drop Tables
	APP_ACTION_EXECUTION (ID NUMERIC(19) NOT NULL, END_TIME DATETIME NULL, LABEL INTEGER NOT NULL, SEQ_ID INTEGER NULL, START_TIME DATETIME NULL, STATUS NVARCHAR(255) NULL, TASK_IDS NVARCHAR(255) NULL, VERSIONLOCK NUMERIC(19) NULL, KAPPLET_RUN_ID NUMERIC(19) NOT NULL, PRIMARY KEY (ID)); CREATE TABLE APP_DATA_ROW (ID NUMERIC(19) NOT NULL, TABLE_ID NUMERIC(19) NOT NULL, PRIMARY KEY (ID)); CREATE TABLE APP_DATA_TABLE (ID NUMERIC(19) NOT NULL, TABLE_NAME NVARCHAR(255) NOT NULL, FLOW_EXECUTION_ID NUMERIC(19) NOT NULL, PRIMARY KEY (ID)); CREATE TABLE APP_DATA_STR (ID NUMERIC(19) NOT NULL, NAME NVARCHAR(255) NULL, VALUE NVARCHAR(2000) NOT NULL, RESULT_ID NUMERIC(19) NOT NULL, PRIMARY KEY (ID)); CREATE TABLE APP_DATA_LOB (ID NUMERIC(19) NOT NULL, LARGE BIT default 0 NULL, NAME NVARCHAR(255) NULL, VALUE NTEXT NULL, RESULT_ID NUMERIC(19) NOT NULL, PRIMARY KEY (ID)); CREATE TABLE APP_FLOW_ERROR (ID NUMERIC(19) IDENTITY NOT NULL, ERROR NTEXT NULL, EXECUTION_ID NUMERIC(19) NOT NULL, PRIMARY KEY (ID)); CREATE TABLE APP_RUN (ID NUMERIC(19) IDENTITY NOT NULL, ABORTED BIT default 0 NULL, FLOW NTEXT NULL, INITIAL_LABEL INTEGER NOT NULL, LASTEXECUTIONPATHCHANGE NUMERIC(19) NULL, OPTIONS NTEXT NULL, PARAMETERS NTEXT NULL, SIGNATURES NTEXT NULL, VERSIONLOCK NUMERIC(19) NULL, KAPPLET_ID NVARCHAR(40) NOT NULL, PRIMARY KEY (ID)); CREATE TABLE APP_RUN_VALUE (ID NVARCHAR(255) NOT NULL, VALUE IMAGE NOT NULL, RUN_ID NUMERIC(19) NOT NULL, PRIMARY KEY (ID)); CREATE TABLE MC_FOLDER (ID NUMERIC(19) IDENTITY NOT NULL, DTYPE NVARCHAR(31) NULL, PATH NVARCHAR(255) NULL, PRIMARY KEY (ID)); CREATE TABLE MC_ROOT_FOLDER (ID NUMERIC(19) NOT NULL, P_ID NUMERIC(19) NOT NULL, PRIMARY KEY (ID)); CREATE TABLE MC_SUBFOLDER (ID NUMERIC(19) NOT NULL, NAME NVARCHAR(255) NOT NULL, PARENT_ID NUMERIC(19) NOT NULL, PRIMARY KEY (ID)); CREATE TABLE APP_RUN_APP_SCHEDULE (schedules_ID NUMERIC(19) NOT NULL, kappletRuns_ID NUMERIC(19) NOT NULL, PRIMARY KEY (schedules_ID, kappletRuns_ID)); ALTER TABLE MC_SCHEDULE ADD CONSTRAINT UNQ_MC_SCHEDULE_0 UNIQUE (P_ID, NAME); ALTER TABLE MC_BLOCK ADD CONSTRAINT UNQ_MC_BLOCK_0 UNIQUE (P_ID, NAME); ALTER TABLE MC_BLOCK_REFERENCE ADD CONSTRAINT MC_BLOCK_REFERENCE0 UNIQUE (PARENT, NAME); ALTER TABLE MC_ROBOT ADD CONSTRAINT UNQ_MC_ROBOT_0 UNIQUE (P_ID, NAME, FOLDER_EX); ALTER TABLE MC_TYPE ADD CONSTRAINT UNQ_MC_TYPE_0 UNIQUE (P_ID,	APP_ACTION_EXECUTION DROP CONSTRAINT APP_ACTION_EXECUTION0; ALTER TABLE APP_DATA_ROW DROP CONSTRAINT APP_DATA_ROW_TABLE_ID; ALTER TABLE APP_DATA_TABLE DROP CONSTRAINT PPDTTABLEFLWXECUTIONID; ALTER TABLE APP_DATA_TABLE DROP CONSTRAINT UNQ_APP_DATA_TABLE_0; ALTER TABLE APP_DATA_STR DROP CONSTRAINT APP_DATA_STR_RESULT_ID; ALTER TABLE APP_DATA_LOB DROP CONSTRAINT APP_DATA_LOB_RESULT_ID; ALTER TABLE APP_FLOW_ERROR DROP CONSTRAINT PPFLOWERROREXECUTIONID; ALTER TABLE APP_RUN DROP CONSTRAINT FK_APP_RUN_KAPPLET_ID; ALTER TABLE APP_RUN_VALUE DROP CONSTRAINT APP_RUN_VALUE_RUN_ID; ALTER TABLE MC_ROOT_FOLDER DROP CONSTRAINT FK_MC_ROOT_FOLDER_P_ID; ALTER TABLE MC_ROOT_FOLDER DROP CONSTRAINT FK_MC_ROOT_FOLDER_ID; ALTER TABLE MC_SUBFOLDER DROP CONSTRAINT FK_MC_SUBFOLDER_ID; ALTER TABLE MC_SUBFOLDER DROP CONSTRAINT MC_SUBFOLDER_PARENT_ID; ALTER TABLE MC_SUBFOLDER DROP CONSTRAINT UNQ_MC_SUBFOLDER_0; ALTER TABLE APP_RUN_APP_SCHEDULE DROP CONSTRAINT

Database	Create Tables	Drop Tables
	NAME); ALTER TABLE MC_SNIPPET ADD CONSTRAINT UNQ_MC_SNIPPET_0 UNIQUE (P_ID, NAME); ALTER TABLE MC_RESOURCE_FILE ADD CONSTRAINT UNQ_MC_RESOURCE_FILE_0 UNIQUE (P_ID, NAME, FOLDER_EX); ALTER TABLE MC_GROUP_TO_ROLE_MAP ADD CONSTRAINT MC_GROUP_TO_ROLE_MAP0 UNIQUE (P_ID, ROLE_NAME, GROUP_NAME); ALTER TABLE MC_OAUTH_APPLICATION ADD CONSTRAINT MC_OAUTH_APPLICATION0 UNIQUE (P_ID, NAME); ALTER TABLE MC_OAUTH_USER ADD CONSTRAINT UNQ_MC_OAUTH_USER_0 UNIQUE (APPLICATION_ID, NAME); ALTER TABLE MC_GROUP ADD CONSTRAINT UNQ_MC_GROUP_0 UNIQUE (GROUP_NAME); ALTER TABLE MC_USER_GROUP_REL ADD CONSTRAINT MC_USER_GROUP_REL0 UNIQUE (USER_ID, GROUP_ID); ALTER TABLE MC_USER_PROP ADD CONSTRAINT UNQ_MC_USER_PROP_0 UNIQUE (ID, PROP_NAME); ALTER TABLE MC_KAPPLET_PROP ADD CONSTRAINT UNQ_MC_KAPPLET_PROP_0 UNIQUE (ID, PROP_NAME); ALTER TABLE APP_ACTION_EXECUTION ADD CONSTRAINT APP_ACTION_EXECUTION0 UNIQUE (KAPPLET_RUN_ID, LABEL); ALTER TABLE APP_DATA_TABLE ADD CONSTRAINT UNQ_APP_DATA_TABLE_0 UNIQUE (FLOW_EXECUTION_ID, TABLE_NAME); ALTER TABLE MC_SUBFOLDER ADD CONSTRAINT UNQ_MC_SUBFOLDER_0 UNIQUE (NAME, PARENT_ID); ALTER TABLE MC_SCHEDULE ADD CONSTRAINT MC_SCHEDULEBLOCKJOB_ID FOREIGN KEY (BLOCKJOB_ID) REFERENCES MC_BLOCK_JOB (ID); ALTER TABLE MC_SCHEDULE ADD CONSTRAINT FK_MC_SCHEDULE_P_ID FOREIGN KEY (P_ID) REFERENCES MC_PROJECT (ID); ALTER TABLE MC_SCHEDULE ADD CONSTRAINT MC_SCHEDULE_CLUSTER_ID FOREIGN KEY (CLUSTER_ID) REFERENCES MC_ROBOSERVER_CLUSTER (ID); ALTER TABLE MC_BLOCK ADD CONSTRAINT FK_MC_BLOCK_P_ID FOREIGN KEY (P_ID) REFERENCES MC_PROJECT (ID); ALTER TABLE MC_BLOCK_REFERENCE ADD CONSTRAINT MCBLOCK_REFERENCECHILD FOREIGN KEY (CHILD) REFERENCES MC_BLOCK (ID); ALTER TABLE MC_BLOCK_REFERENCE ADD CONSTRAINT MCBLOCKREFERENCEPARENT FOREIGN KEY (PARENT) REFERENCES MC_BLOCK (ID); ALTER TABLE MC_BLOCK_JOB ADD CONSTRAINT MC_BLOCK_JOB_BLOCK_ID FOREIGN KEY (BLOCK_ID) REFERENCES MC_BLOCK (ID); ALTER TABLE MC_ROBOT_BLOCK ADD CONSTRAINT FK_MC_ROBOT_BLOCK_ID FOREIGN KEY (ID) REFERENCES MC_BLOCK (ID); ALTER TABLE MC_ROBOT_BLOCK ADD CONSTRAINT MC_ROBOT_BLOCKROBOT_ID FOREIGN KEY (ROBOT_ID) REFERENCES MC_FOLDER_ITEM (ID); ALTER TABLE MC_MULTIPLE_ROBOT_BLOCK ADD CONSTRAINT MCMULTIPLEROBOTBLOCKID FOREIGN KEY (ID) REFERENCES MC_BLOCK (ID); ALTER TABLE MC_CONTAINER_BLOCK	PPRNPPSCHEDULEschdlsID; ALTER TABLE APP_RUN_APP_SCHEDULE DROP CONSTRAINT PPRNPPSCHDULEkppltRnsD; DROP TABLE MC_SCHEDULE; DROP TABLE MC_BLOCK; DROP TABLE MC_BLOCK_REFERENCE; DROP TABLE MC_BLOCK_JOB; DROP TABLE MC_ROBOT_BLOCK; DROP TABLE MC_MULTIPLE_ROBOT_BLOCK; DROP TABLE MC_CONTAINER_BLOCK; DROP TABLE MC_SEQUENTIAL_BLOCK; DROP TABLE MC_CONCURRENT_BLOCK; DROP TABLE MC_SCRIPT_BLOCK; DROP TABLE MC_FOLDER_ITEM; DROP TABLE MC_ROBOT; DROP TABLE MC_TYPE; DROP TABLE MC_SNIPPET; DROP TABLE MC_ROBOSERVER; DROP TABLE MC_SETTINGS; DROP TABLE MC_ROBOT_PARAMETER; DROP TABLE MC_ROBOT_REF_TYPE; DROP TABLE MC_ROBOT_REF_SNIPPET; DROP TABLE MC_ROBOT_TYPED_VARIABLE; DROP TABLE MC_ROBOT_REF_VAR; DROP TABLE MC_RESOURCE_FILE; DROP TABLE MC_PROJECT; DROP TABLE MC_ROBOSERVER_CLUSTER; DROP TABLE MC_GROUP_TO_ROLE_MAP; DROP TABLE MC_SNIPPET_PARAMETER; DROP TABLE MC_SNIPPET_REF_SNIPPET; DROP TABLE

Database	Create Tables	Drop Tables
	ADD CONSTRAINT MC_CONTAINER_BLOCK_ID FOREIGN KEY (ID) REFERENCES MC_BLOCK (ID); ALTER TABLE MC_SEQUENTIAL_BLOCK ADD CONSTRAINT MC_SEQUENTIAL_BLOCK_ID FOREIGN KEY (ID) REFERENCES MC_BLOCK (ID); ALTER TABLE MC_CONCURRENT_BLOCK ADD CONSTRAINT MC_CONCURRENT_BLOCK_ID FOREIGN KEY (ID) REFERENCES MC_BLOCK (ID); ALTER TABLE MC_SCRIPT_BLOCK ADD CONSTRAINT FK_MC_SCRIPT_BLOCK_ID FOREIGN KEY (ID) REFERENCES MC_BLOCK (ID); ALTER TABLE MC_FOLDER_ITEM ADD CONSTRAINT MC_FOLDER_ITEM_FOLDER FOREIGN KEY (FOLDER) REFERENCES MC_FOLDER (ID); ALTER TABLE MC_ROBOT ADD CONSTRAINT FK_MC_ROBOT_FOLDER_EX FOREIGN KEY (FOLDER_EX) REFERENCES MC_FOLDER (ID); ALTER TABLE MC_ROBOT ADD CONSTRAINT FK_MC_ROBOT_ID FOREIGN KEY (ID) REFERENCES MC_FOLDER_ITEM (ID); ALTER TABLE MC_ROBOT ADD CONSTRAINT FK_MC_ROBOT_P_ID FOREIGN KEY (P_ID) REFERENCES MC_PROJECT (ID); ALTER TABLE MC_TYPE ADD CONSTRAINT FK_MC_TYPE_P_ID FOREIGN KEY (P_ID) REFERENCES MC_PROJECT (ID); ALTER TABLE MC_TYPE ADD CONSTRAINT FK_MC_TYPE_ID FOREIGN KEY (ID) REFERENCES MC_FOLDER_ITEM (ID); ALTER TABLE MC_SNIPPET ADD CONSTRAINT FK_MC_SNIPPET_P_ID FOREIGN KEY (P_ID) REFERENCES MC_PROJECT (ID); ALTER TABLE MC_SNIPPET ADD CONSTRAINT FK_MC_SNIPPET_ID FOREIGN KEY (ID) REFERENCES MC_FOLDER_ITEM (ID); ALTER TABLE MC_ROBOSERVER ADD CONSTRAINT MCROBOSERVERCLUSTER_ID FOREIGN KEY (CLUSTER_ID) REFERENCES MC_ROBOSERVER_CLUSTER (ID); ALTER TABLE MC_ROBOT_PARAMETER ADD CONSTRAINT MCROBOTPARAMETERRBOTID FOREIGN KEY (ROBOT_ID) REFERENCES MC_FOLDER_ITEM (ID); ALTER TABLE MC_ROBOT_REF_TYPE ADD CONSTRAINT MCROBOTREFTYPEROBOT_ID FOREIGN KEY (ROBOT_ID) REFERENCES MC_FOLDER_ITEM (ID); ALTER TABLE MC_ROBOT_REF_SNIPPET ADD CONSTRAINT MCRBOTREFSNIPPETRBOTID FOREIGN KEY (ROBOT_ID) REFERENCES MC_FOLDER_ITEM (ID); ALTER TABLE MC_ROBOT_TYPED_VARIABLE ADD CONSTRAINT MCRBTTYPEDVARIABLERBTD FOREIGN KEY (ROBOT_ID) REFERENCES MC_FOLDER_ITEM (ID); ALTER TABLE MC_ROBOT_REF_VAR ADD CONSTRAINT MCROBOTREF_VARROBOT_ID FOREIGN KEY (ROBOT_ID) REFERENCES MC_FOLDER_ITEM (ID); ALTER TABLE MC_RESOURCE_FILE ADD CONSTRAINT MCRESOURCEFILEFOLDEREX FOREIGN KEY (FOLDER_EX) REFERENCES MC_FOLDER (ID); ALTER TABLE MC_RESOURCE_FILE ADD CONSTRAINT MC_RESOURCE_FILE_P_ID FOREIGN KEY (P_ID) REFERENCES MC_PROJECT	MC_SNIPPET_REF_TYPE; DROP TABLE MC_SNIPPET_TYPED_VAR; DROP TABLE MC_SNIPPET_REF_VAR; DROP TABLE MC_OAUTH_APPLICATION; DROP TABLE MC_OAUTH_USER; DROP TABLE MC_DSLICENSE; DROP TABLE MC_USER; DROP TABLE MC_GROUP; DROP TABLE MC_USER_GROUP_REL; DROP TABLE MC_USER_PROP; DROP TABLE MC_TYPE_ATTR; DROP TABLE MC_RS_CLUSTER_SETTINGS; DROP TABLE MC_DATABASE; DROP TABLE MC_PROXY_SERVER; DROP TABLE MC_DATABASE_TYPE; DROP TABLE MC_JAR_FILE; DROP TABLE APP_KAPPLET; DROP TABLE APP_MASTER; DROP TABLE APP_INSTALLED; DROP TABLE APP_ICON; DROP TABLE APP_UPLOAD; DROP TABLE MC_KAPPLET_PROP; DROP TABLE APP_SCHEDULE_ERROR; DROP TABLE APP_SCHEDULE; DROP TABLE APP_SCHEDULE_VALUE; DROP TABLE MC_GLOBAL_KAPPLET_PROP; DROP TABLE APP_ACTION_EXECUTION; DROP TABLE APP_DATA_ROW; DROP TABLE APP_DATA_TABLE; DROP TABLE APP_DATA_STR; DROP TABLE APP_DATA_LOB;

Database	Create Tables	Drop Tables
	(ID); ALTER TABLE MC_RESOURCE_FILE ADD CONSTRAINT FK_MC_RESOURCE_FILE_ID FOREIGN KEY (ID) REFERENCES MC_FOLDER_ITEM (ID); ALTER TABLE MC_PROJECT ADD CONSTRAINT MC_PROJECT_CLUSTER_ID FOREIGN KEY (CLUSTER_ID) REFERENCES MC_ROBOSERVER_CLUSTER (ID); ALTER TABLE MC_ROBOSERVER_CLUSTER ADD CONSTRAINT MCRBSRVRCLUSTERSTTNGSD FOREIGN KEY (SETTINGS_ID) REFERENCES MC_RS_CLUSTER_SETTINGS (ID); ALTER TABLE MC_GROUP_TO_ROLE_MAP ADD CONSTRAINT MCGROUPTO_ROLE_MAPP_ID FOREIGN KEY (P_ID) REFERENCES MC_PROJECT (ID); ALTER TABLE MC_SNIPPET_PARAMETER ADD CONSTRAINT MCSNPPTPARAMETERSNPPTD FOREIGN KEY (SNIPPET_ID) REFERENCES MC_FOLDER_ITEM (ID); ALTER TABLE MC_SNIPPET_REF_SNIPPET ADD CONSTRAINT MCSNPPTRFSNIPPETSNPPTD FOREIGN KEY (SNIPPET_ID) REFERENCES MC_FOLDER_ITEM (ID); ALTER TABLE MC_SNIPPET_REF_TYPE ADD CONSTRAINT MCSNPPETREFTYPESNPPTID FOREIGN KEY (SNIPPET_ID) REFERENCES MC_FOLDER_ITEM (ID); ALTER TABLE MC_SNIPPET_TYPED_VAR ADD CONSTRAINT MCSNPPTTYPEDVARSNPPTID FOREIGN KEY (SNIPPET_ID) REFERENCES MC_FOLDER_ITEM (ID); ALTER TABLE MC_SNIPPET_REF_VAR ADD CONSTRAINT MCSNPPETREFVARSNPPETID FOREIGN KEY (SNIPPET_ID) REFERENCES MC_FOLDER_ITEM (ID); ALTER TABLE MC_OAUTH_APPLICATION ADD CONSTRAINT MCOAUTHAPPLICATIONP_ID FOREIGN KEY (P_ID) REFERENCES MC_PROJECT (ID); ALTER TABLE MC_OAUTH_USER ADD CONSTRAINT MCAUTHUSERPPLICATIONID FOREIGN KEY (APPLICATION_ID) REFERENCES MC_OAUTH_APPLICATION (ID); ALTER TABLE MC_OAUTH_USER ADD CONSTRAINT MC_OAUTH_USER_USER_ID FOREIGN KEY (USER_ID) REFERENCES MC_USER (USER_NAME); ALTER TABLE MC_USER_GROUP_REL ADD CONSTRAINT MCUSERGROUP_RELUSER_ID FOREIGN KEY (USER_ID) REFERENCES MC_USER (USER_NAME); ALTER TABLE MC_USER_GROUP_REL ADD CONSTRAINT MCUSERGROUPRELGROUP_ID FOREIGN KEY (GROUP_ID) REFERENCES MC_GROUP (ID); ALTER TABLE MC_USER_PROP ADD CONSTRAINT MC_USER_PROP_USER_ID FOREIGN KEY (USER_ID) REFERENCES MC_USER (USER_NAME); ALTER TABLE MC_TYPE_ATTR ADD CONSTRAINT MC_TYPE_ATTR_TYPE_ID FOREIGN KEY (TYPE_ID) REFERENCES MC_FOLDER_ITEM (ID); ALTER TABLE MC_DATABASE ADD CONSTRAINT MCDTBSECLSTRSETTINGSID FOREIGN KEY (CLUSTER_SETTINGS_ID) REFERENCES MC_RS_CLUSTER_SETTINGS (ID); ALTER TABLE MC_PROXY_SERVER ADD CONSTRAINT MCPRXYSRVRCLSTRSTTNGSD FOREIGN KEY (CLUSTER_SETTINGS_ID)	DROP TABLE APP_FLOW_ERROR; DROP TABLE APP_RUN; DROP TABLE APP_RUN_VALUE; DROP TABLE MC_FOLDER; DROP TABLE MC_ROOT_FOLDER; DROP TABLE MC_SUBFOLDER; DROP TABLE APP_RUN_APP_SCHEDULE; DROP TABLE APP_SEQUENCES;

Database	Create Tables	Drop Tables
	REFERENCES MC_RS_CLUSTER_SETTINGS (ID); ALTER TABLE APP_KAPPLET ADD CONSTRAINT FK_APP_KAPPLET_ICON_ID FOREIGN KEY (ICON_ID) REFERENCES APP_ICON (ID); ALTER TABLE APP_KAPPLET ADD CONSTRAINT FK_APP_KAPPLET_P_ID FOREIGN KEY (P_ID) REFERENCES MC_PROJECT (ID); ALTER TABLE APP_MASTER ADD CONSTRAINT FK_APP_MASTER_ID FOREIGN KEY (ID) REFERENCES APP_KAPPLET (ID); ALTER TABLE APP_INSTALLED ADD CONSTRAINT APP_INSTALLED_USER_ID FOREIGN KEY (USER_ID) REFERENCES MC_USER (USER_NAME); ALTER TABLE APP_INSTALLED ADD CONSTRAINT APP_INSTALLED_PARENT FOREIGN KEY (PARENT) REFERENCES APP_KAPPLET (ID); ALTER TABLE APP_INSTALLED ADD CONSTRAINT FK_APP_INSTALLED_ID FOREIGN KEY (ID) REFERENCES APP_KAPPLET (ID); ALTER TABLE MC_KAPPLET_PROP ADD CONSTRAINT MCKAPPLETPROPKAPPLETID FOREIGN KEY (KAPPLET_ID) REFERENCES APP_KAPPLET (ID); ALTER TABLE APP_SCHEDULE_ERROR ADD CONSTRAINT PPSCHDULEERRORSCHDLEID FOREIGN KEY (SCHEDULE_ID) REFERENCES APP_SCHEDULE (ID); ALTER TABLE APP_SCHEDULE ADD CONSTRAINT FK_APP_SCHEDULE_APP_ID FOREIGN KEY (APP_ID) REFERENCES APP_KAPPLET (ID); ALTER TABLE APP_SCHEDULE_VALUE ADD CONSTRAINT PPSCHDULEVALUESCHDLEID FOREIGN KEY (SCHEDULE_ID) REFERENCES APP_SCHEDULE (ID); ALTER TABLE APP_ACTION_EXECUTION ADD CONSTRAINT PPCTNXECUTIONKPPLTRNID FOREIGN KEY (KAPPLET_RUN_ID) REFERENCES APP_RUN (ID); ALTER TABLE APP_DATA_ROW ADD CONSTRAINT APP_DATA_ROW_TABLE_ID FOREIGN KEY (TABLE_ID) REFERENCES APP_DATA_TABLE (ID); ALTER TABLE APP_DATA_TABLE ADD CONSTRAINT PPDTTABLEFLWXECUTIONID FOREIGN KEY (FLOW_EXECUTION_ID) REFERENCES APP_ACTION_EXECUTION (ID); ALTER TABLE APP_DATA_STR ADD CONSTRAINT APP_DATA_STR_RESULT_ID FOREIGN KEY (RESULT_ID) REFERENCES APP_DATA_ROW (ID); ALTER TABLE APP_DATA_LOB ADD CONSTRAINT APP_DATA_LOB_RESULT_ID FOREIGN KEY (RESULT_ID) REFERENCES APP_DATA_ROW (ID); ALTER TABLE APP_FLOW_ERROR ADD CONSTRAINT PPFLOWERROREXECUTIONID FOREIGN KEY (EXECUTION_ID) REFERENCES APP_ACTION_EXECUTION (ID); ALTER TABLE APP_RUN ADD CONSTRAINT FK_APP_RUN_KAPPLET_ID FOREIGN KEY (KAPPLET_ID) REFERENCES APP_KAPPLET (ID); ALTER TABLE APP_RUN_VALUE ADD CONSTRAINT APP_RUN_VALUE_RUN_ID FOREIGN KEY (RUN_ID) REFERENCES APP_RUN (ID); ALTER TABLE MC_ROOT_FOLDER ADD CONSTRAINT FK_MC_ROOT_FOLDER_P_ID FOREIGN KEY (P_ID) REFERENCES MC_PROJECT	

Database	Create Tables	Drop Tables
	(ID); ALTER TABLE MC_ROOT_FOLDER ADD CONSTRAINT FK_MC_ROOT_FOLDER_ID FOREIGN KEY (ID) REFERENCES MC_FOLDER (ID); ALTER TABLE MC_SUBFOLDER ADD CONSTRAINT FK_MC_SUBFOLDER_ID FOREIGN KEY (ID) REFERENCES MC_FOLDER (ID); ALTER TABLE MC_SUBFOLDER ADD CONSTRAINT MC_SUBFOLDER_PARENT_ID FOREIGN KEY (PARENT_ID) REFERENCES MC_FOLDER (ID); ALTER TABLE APP_RUN_APP_SCHEDULE ADD CONSTRAINT PPRNPPSCHEDULEschdlsID FOREIGN KEY (schedules_ID) REFERENCES APP_SCHEDULE (ID); ALTER TABLE APP_RUN_APP_SCHEDULE ADD CONSTRAINT PPRNPPSCHDULEkppltRnsD FOREIGN KEY (kappletRuns_ID) REFERENCES APP_RUN (ID); CREATE TABLE APP_SEQUENCES (TABLE_NAME NVARCHAR(50) NOT NULL, NEXT_ID NUMERIC(28) NULL, PRIMARY KEY (TABLE_NAME)); INSERT INTO APP_SEQUENCES(TABLE_NAME, NEXT_ID) values ('DATA_STR', 0); INSERT INTO APP_SEQUENCES(TABLE_NAME, NEXT_ID) values ('DATA_ROW', 0); INSERT INTO APP_SEQUENCES(TABLE_NAME, NEXT_ID) values ('DATA_TABLE', 0); INSERT INTO APP_SEQUENCES(TABLE_NAME, NEXT_ID) values ('DATA_LOB', 0); INSERT INTO APP_SEQUENCES(TABLE_NAME, NEXT_ID) values ('FLOW_ACTION_EXECUTION', 0);	
Sybase	CREATE TABLE MC_SCHEDULE (ID NUMERIC(19) NOT NULL, BAD_INPUT BIT default 0 NOT NULL, ACTIVE BIT default 0 NOT NULL, CREATEDBY NVARCHAR(255) NULL, DIRTY BIT default 0 NOT NULL, EMAILS NVARCHAR(255) NULL, MAXOBJECTSEXTRACTED INTEGER NULL, MAXRUNTIME INTEGER NULL, MODIFIEDBY NVARCHAR(255) NULL, NAME NVARCHAR(255) NOT NULL, NEXTRUN DATETIME NULL, PREVIOUSRUN DATETIME NULL, TOTALRUNS INTEGER NULL, USEEMAILNOTIFICATION BIT default 0 NOT NULL, CLUSTER_ID NUMERIC(19) NULL, P_ID NUMERIC(19) NOT NULL, BLOCKJOB_ID NUMERIC(19) NULL, PRIMARY KEY (ID)); CREATE TABLE MC_BLOCK (ID NUMERIC(19) IDENTITY NOT NULL, DTYPE NVARCHAR(31) NULL, NAME NVARCHAR(255) NOT NULL, NAMEDBLOCK BIT default 0 NOT NULL, P_ID NUMERIC(19) NOT NULL, PRIMARY KEY (ID)); CREATE TABLE MC_BLOCK_REFERENCE (ID NUMERIC(19) IDENTITY NOT NULL, ENABLED BIT default 0 NOT NULL, ORDER_VAL INTEGER NOT NULL, NAME NVARCHAR(255) NOT NULL, CHILD NUMERIC(19) NULL, PARENT NUMERIC(19) NOT NULL, PRIMARY KEY (ID)); CREATE TABLE MC_BLOCK_JOB (ID NUMERIC(19) IDENTITY NOT NULL, ACTIVE BIT default 0 NOT NULL, BLOCKINPUT TEXT NULL, DISPLAY_NAMES TEXT NOT NULL, BLOCK_ID NUMERIC(19) NOT NULL, PRIMARY KEY (ID)); CREATE TABLE	ALTER TABLE MC_SCHEDULE DROP CONSTRAINT MC_SCHEDULEBLOCKJOB_ID; ALTER TABLE MC_SCHEDULE DROP CONSTRAINT FK_MC_SCHEDULE_P_ID; ALTER TABLE MC_SCHEDULE DROP CONSTRAINT MC_SCHEDULE_CLUSTER_ID; ALTER TABLE MC_SCHEDULE DROP CONSTRAINT UNQ_MC_SCHEDULE_0; ALTER TABLE MC_BLOCK DROP CONSTRAINT FK_MC_BLOCK_P_ID; ALTER TABLE MC_BLOCK DROP CONSTRAINT UNQ_MC_BLOCK_0; ALTER TABLE MC_BLOCK_REFERENCE DROP CONSTRAINT MCBLOCK_REFERENCECHILD; ALTER TABLE

Database	Create Tables	Drop Tables
	MC_ROBOT_BLOCK (ID NUMERIC(19) NOT NULL, ROBOT_ID NUMERIC(19) NULL, PRIMARY KEY (ID)); CREATE TABLE MC_MULTIPLE_ROBOT_BLOCK (ID NUMERIC(19) NOT NULL, DISPLAY_NAME NVARCHAR(255) NULL, STRATEGY NVARCHAR(255) NULL, VALUE NVARCHAR(255) NULL, PRIMARY KEY (ID)); CREATE TABLE MC_CONTAINER_BLOCK (ID NUMERIC(19) NOT NULL, MAPPING IMAGE NULL, PRIMARY KEY (ID)); CREATE TABLE MC_SEQUENTIAL_BLOCK (ID NUMERIC(19) NOT NULL, PRIMARY KEY (ID)); CREATE TABLE MC_CONCURRENT_BLOCK (ID NUMERIC(19) NOT NULL, PRIMARY KEY (ID)); CREATE TABLE MC_SCRIPT_BLOCK (ID NUMERIC(19) NOT NULL, SCRIPT NVARCHAR(255) NULL, PRIMARY KEY (ID)); CREATE TABLE MC_FOLDER_ITEM (ID NUMERIC(19) IDENTITY NOT NULL, DTYPE NVARCHAR(31) NULL, FOLDER NUMERIC(19) NOT NULL, PRIMARY KEY (ID)); CREATE TABLE MC_ROBOT (ID NUMERIC(19) NOT NULL, BYTESIZE INTEGER NOT NULL, CONTAINSRETURNOBJECT BIT default 0 NOT NULL, CREATEDBY NVARCHAR(255) NULL, LASTMODIFIED DATETIME NOT NULL, MODIFIEDBY NVARCHAR(255) NULL, NAME NVARCHAR(255) NOT NULL, ROBOTBYTES IMAGE NOT NULL, SHA_HASH NVARCHAR(255) NOT NULL, VERSION NVARCHAR(255) NOT NULL, FOLDER_EX NUMERIC(19) NOT NULL, P_ID NUMERIC(19) NOT NULL, PRIMARY KEY (ID)); CREATE TABLE MC_TYPE (ID NUMERIC(19) NOT NULL, CREATEDBY NVARCHAR(255) NULL, LASTMODIFIED DATETIME NULL, MODIFIEDBY NVARCHAR(255) NULL, NAME NVARCHAR(255) NOT NULL, SHA_HASH NVARCHAR(255) NOT NULL, TYPEBYTES IMAGE NOT NULL, P_ID NUMERIC(19) NOT NULL, PRIMARY KEY (ID)); CREATE TABLE MC_SNIPPET (ID NUMERIC(19) NOT NULL, CREATEDBY NVARCHAR(255) NULL, LASTMODIFIED DATETIME NOT NULL, MODIFIEDBY NVARCHAR(255) NULL, NAME NVARCHAR(255) NOT NULL, SHA_HASH NVARCHAR(255) NOT NULL, SNIPPETBYTES IMAGE NOT NULL, VERSION NVARCHAR(255) NOT NULL, P_ID NUMERIC(19) NOT NULL, PRIMARY KEY (ID)); CREATE TABLE MC_ROBOSERVER (ID NUMERIC(19) IDENTITY NOT NULL, HOST NVARCHAR(255) NOT NULL, PORT INTEGER NOT NULL, CLUSTER_ID NUMERIC(19) NOT NULL, PRIMARY KEY (ID)); CREATE TABLE MC_SETTINGS (NAME NVARCHAR(255) NOT NULL, VALUE NVARCHAR(255) NULL, PRIMARY KEY (NAME)); CREATE TABLE MC_ROBOT_PARAMETER (ID NUMERIC(19) IDENTITY NOT NULL, TYPENAME NVARCHAR(255) NOT NULL, VARIABLENAME NVARCHAR(255) NOT NULL, ROBOT_ID NUMERIC(19) NOT NULL, PRIMARY KEY (ID)); CREATE TABLE MC_ROBOT_REF_TYPE (ID NUMERIC(19) IDENTITY NOT NULL, ISINPUT BIT default 0 NOT NULL, ISRETURNED BIT default 0 NOT	MC_BLOCK_REFERENCE DROP CONSTRAINT MCBLOCKREFERENCEPARENT; ALTER TABLE MC_BLOCK_REFERENCE DROP CONSTRAINT MC_BLOCK_REFERENCE0; ALTER TABLE MC_BLOCK_JOB DROP CONSTRAINT MC_BLOCK_JOB_BLOCK_ID; ALTER TABLE MC_ROBOT_BLOCK DROP CONSTRAINT FK_MC_ROBOT_BLOCK_ID; ALTER TABLE MC_ROBOT_BLOCK DROP CONSTRAINT MC_ROBOT_BLOCKROBOT_ID; ALTER TABLE MC_MULTIPLE_ROBOT_BLOCK DROP CONSTRAINT MCMULTIPLEROBOTBLOCKID; ALTER TABLE MC_CONTAINER_BLOCK DROP CONSTRAINT MC_CONTAINER_BLOCK_ID; ALTER TABLE MC_SEQUENTIAL_BLOCK DROP CONSTRAINT MC_SEQUENTIAL_BLOCK_ID; ALTER TABLE MC_CONCURRENT_BLOCK DROP CONSTRAINT MC_CONCURRENT_BLOCK_ID; ALTER TABLE MC_SCRIPT_BLOCK DROP CONSTRAINT FK_MC_SCRIPT_BLOCK_ID; ALTER TABLE MC_FOLDER_ITEM DROP CONSTRAINT MC_FOLDER_ITEM_FOLDER; ALTER TABLE MC_ROBOT DROP CONSTRAINT FK_MC_ROBOT_FOLDER_EX; ALTER TABLE MC_ROBOT DROP CONSTRAINT FK_MC_ROBOT_ID; ALTER TABLE MC_ROBOT DROP CONSTRAINT FK_MC_ROBOT_P_ID; ALTER TABLE MC_ROBOT DROP CONSTRAINT

Database	Create Tables	Drop Tables
	NULL, ISSTORED BIT default 0 NOT NULL, NAME NVARCHAR(255) NOT NULL, ROBOT_ID NUMERIC(19) NOT NULL, PRIMARY KEY (ID)); CREATE TABLE MC_ROBOT_REF_SNIPPET (ID NUMERIC(19) IDENTITY NOT NULL, NAME NVARCHAR(255) NOT NULL, ROBOT_ID NUMERIC(19) NOT NULL, PRIMARY KEY (ID)); CREATE TABLE MC_ROBOT_TYPED_VARIABLE (ID NUMERIC(19) IDENTITY NOT NULL, TYPENAME NVARCHAR(255) NOT NULL, VARIABLENAME NVARCHAR(255) NOT NULL, ROBOT_ID NUMERIC(19) NOT NULL, PRIMARY KEY (ID)); CREATE TABLE MC_ROBOT_REF_VAR (ID NUMERIC(19) IDENTITY NOT NULL, ISRETURNED BIT default 0 NOT NULL, ISSTORED BIT default 0 NOT NULL, NAME NVARCHAR(255) NULL, ROBOT_ID NUMERIC(19) NOT NULL, PRIMARY KEY (ID)); CREATE TABLE MC_RESOURCE_FILE (ID NUMERIC(19) NOT NULL, BYTESIZE INTEGER NOT NULL, BYTES IMAGE NOT NULL, CREATEDBY NVARCHAR(255) NULL, LASTMODIFIED DATETIME NOT NULL, MODIFIEDBY NVARCHAR(255) NULL, NAME NVARCHAR(255) NOT NULL, SHA_HASH NVARCHAR(255) NOT NULL, FOLDER_EX NUMERIC(19) NOT NULL, P_ID NUMERIC(19) NOT NULL, PRIMARY KEY (ID)); CREATE TABLE MC_PROJECT (ID NUMERIC(19) IDENTITY NOT NULL, ACAO NVARCHAR(255) NULL, AUTHENTICATE BIT default 0 NOT NULL, DESCRIPTION NVARCHAR(255) NULL, NAME NVARCHAR(255) NOT NULL UNIQUE, CLUSTER_ID NUMERIC(19) NULL, PRIMARY KEY (ID)); CREATE TABLE MC_ROBOSERVER_CLUSTER (ID NUMERIC(19) IDENTITY NOT NULL, KCU INTEGER NULL, NAME NVARCHAR(255) NOT NULL UNIQUE, type BIT default 0 NOT NULL, USESSL BIT default 0 NOT NULL, SETTINGS_ID NUMERIC(19) NULL, PRIMARY KEY (ID)); CREATE TABLE MC_GROUP_TO_ROLE_MAP (ID NUMERIC(19) IDENTITY NOT NULL, GROUP_NAME NVARCHAR(200) NOT NULL, ROLE_NAME NVARCHAR(40) NOT NULL, P_ID NUMERIC(19) NOT NULL, PRIMARY KEY (ID)); CREATE TABLE MC_SNIPPET_PARAMETER (ID NUMERIC(19) IDENTITY NOT NULL, TYPENAME NVARCHAR(255) NOT NULL, VARIABLENAME NVARCHAR(255) NOT NULL, SNIPPET_ID NUMERIC(19) NOT NULL, PRIMARY KEY (ID)); CREATE TABLE MC_SNIPPET_REF_SNIPPET (ID NUMERIC(19) IDENTITY NOT NULL, NAME NVARCHAR(255) NOT NULL, SNIPPET_ID NUMERIC(19) NOT NULL, PRIMARY KEY (ID)); CREATE TABLE MC_SNIPPET_REF_TYPE (ID NUMERIC(19) IDENTITY NOT NULL, ISINPUT BIT default 0 NOT NULL, ISRETURNED BIT default 0 NOT NULL, ISSTORED BIT default 0 NOT NULL, NAME NVARCHAR(255) NULL, SNIPPET_ID NUMERIC(19) NOT NULL, PRIMARY KEY (ID)); CREATE TABLE MC_SNIPPET_TYPED_VAR (ID NUMERIC(19) IDENTITY NOT NULL, TYPENAME NVARCHAR(255)	UNQ_MC_ROBOT_0; ALTER TABLE MC_TYPE DROP CONSTRAINT FK_MC_TYPE_P_ID; ALTER TABLE MC_TYPE DROP CONSTRAINT FK_MC_TYPE_ID; ALTER TABLE MC_TYPE DROP CONSTRAINT UNQ_MC_TYPE_0; ALTER TABLE MC_SNIPPET DROP CONSTRAINT FK_MC_SNIPPET_P_ID; ALTER TABLE MC_SNIPPET DROP CONSTRAINT FK_MC_SNIPPET_ID; ALTER TABLE MC_SNIPPET DROP CONSTRAINT UNQ_MC_SNIPPET_0; ALTER TABLE MC_ROBOSERVER DROP CONSTRAINT MCROBOSERVERCLUSTER_ID; ALTER TABLE MC_ROBOT_PARAMETER DROP CONSTRAINT MCROBOTPARAMETERRBOTID; ALTER TABLE MC_ROBOT_REF_TYPE DROP CONSTRAINT MCROBOTREFTYPEROBOT_ID; ALTER TABLE MC_ROBOT_REF_SNIPPET DROP CONSTRAINT MCRBOTREFSNIPPETRBOTID; ALTER TABLE MC_ROBOT_TYPED_VARIABLE DROP CONSTRAINT MCRBTTYPEDVARIABLERBTD; ALTER TABLE MC_ROBOT_REF_VAR DROP CONSTRAINT MCROBOTREF_VARROBOT_ID; ALTER TABLE MC_RESOURCE_FILE DROP CONSTRAINT MCRESOURCEFILEFOLDEREX; ALTER TABLE MC_RESOURCE_FILE DROP CONSTRAINT MC_RESOURCE_FILE_P_ID; ALTER TABLE

Database	Create Tables	Drop Tables
	NOT NULL, VARIABLENAME NVARCHAR(255) NOT NULL, SNIPPET_ID NUMERIC(19) NOT NULL, PRIMARY KEY (ID)); CREATE TABLE MC_SNIPPET_REF_VAR (ID NUMERIC(19) IDENTITY NOT NULL, ISRETURNED BIT default 0 NOT NULL, ISSTORED BIT default 0 NOT NULL, NAME NVARCHAR(255) NULL, SNIPPET_ID NUMERIC(19) NOT NULL, PRIMARY KEY (ID)); CREATE TABLE MC_OAUTH_APPLICATION (ID NUMERIC(19) IDENTITY NOT NULL, CALLBACKURL NVARCHAR(255) NULL, CONSUMERKEY NVARCHAR(255) NULL, CONSUMERSECRET TEXT NULL, NAME NVARCHAR(255) NOT NULL, SCOPE NVARCHAR(255) NULL, SERVICEPROVIDER NVARCHAR(255) NULL, P_ID NUMERIC(19) NOT NULL, PRIMARY KEY (ID)); CREATE TABLE MC_OAUTH_USER (ID NUMERIC(19) NOT NULL, ACCESSTOKEN NVARCHAR(255) NULL, ACCESSTOKENSECRET TEXT NULL, NAME NVARCHAR(255) NOT NULL, REFRESHTOKEN NVARCHAR(255) NULL, APPLICATION_ID NUMERIC(19) NOT NULL, USER_ID NVARCHAR(255) NULL, PRIMARY KEY (ID)); CREATE TABLE MC_DSLICENSE (TOKEN NVARCHAR(255) NOT NULL, EXPIRATION NUMERIC(19) NULL, IP NVARCHAR(255) NULL, USERNAME NVARCHAR(255) NULL, PRIMARY KEY (TOKEN)); CREATE TABLE MC_USER (USER_NAME NVARCHAR(255) NOT NULL, ADMIN_USER BIT default 0 NOT NULL, CREATION_DATE DATETIME NULL, EMAIL_ADDR NVARCHAR(255) NULL, FULL_NAME NVARCHAR(255) NULL, LAST_IP_ADDR NVARCHAR(255) NULL, LOGIN_TIME DATETIME NULL, LOGIN_COUNT INTEGER NOT NULL, PASS_WORD NVARCHAR(255) NULL, PRIMARY KEY (USER_NAME)); CREATE TABLE MC_GROUP (ID NUMERIC(19) IDENTITY NOT NULL, DESCRIPTION NVARCHAR(255) NULL, GROUP_NAME NVARCHAR(255) NOT NULL, PRIMARY KEY (ID)); CREATE TABLE MC_USER_GROUP_REL (ID NUMERIC(19) IDENTITY NOT NULL, GROUP_ID NUMERIC(19) NOT NULL, USER_ID NVARCHAR(255) NOT NULL, PRIMARY KEY (ID)); CREATE TABLE MC_USER_PROP (ID NUMERIC(19) IDENTITY NOT NULL, IS_NULL BIT default 0 NOT NULL, PROP_NAME NVARCHAR(255) NOT NULL, PROP_VALUE TEXT NULL, USER_ID NVARCHAR(255) NOT NULL, PRIMARY KEY (ID)); CREATE TABLE MC_TYPE_ATTR (ID NUMERIC(19) IDENTITY NOT NULL, ATTR_TYPE NVARCHAR(255) NULL, COMMENTS TEXT NULL, DEFAULT_VALUE TEXT NULL, NAME NVARCHAR(255) NULL, REQUIRED_ATTR BIT default 0 NOT NULL, TYPE_ORDER INTEGER NULL, TYPE_ID NUMERIC(19) NOT NULL, PRIMARY KEY (ID)); CREATE TABLE MC_RS_CLUSTER_SETTINGS (ID NUMERIC(19) IDENTITY NOT NULL, CLUSTER_SETTINGS_ID NVARCHAR(255) NOT NULL, EMAIL_LOG_FROM_ADDRESS NVARCHAR(255) NULL, LOG_INPUT_TO_DB BIT default 0 NOT NULL, LOG_INPUT_TO_LOG4J BIT default 0	MC_RESOURCE_FILE DROP CONSTRAINT FK_MC_RESOURCE_FILE_ID; ALTER TABLE MC_RESOURCE_FILE DROP CONSTRAINT UNQ_MC_RESOURCE_FILE_0; ALTER TABLE MC_PROJECT DROP CONSTRAINT MC_PROJECT_CLUSTER_ID; ALTER TABLE MC_ROBOSERVER_CLUSTER DROP CONSTRAINT MCRBSRVRCLUSTERSTTNGSD; ALTER TABLE MC_GROUP_TO_ROLE_MAP DROP CONSTRAINT MCGROUPTO_ROLE_MAPP_ID; ALTER TABLE MC_GROUP_TO_ROLE_MAP DROP CONSTRAINT MC_GROUP_TO_ROLE_MAP0; ALTER TABLE MC_SNIPPET_PARAMETER DROP CONSTRAINT MCSNPPTPARAMETERSNPPTD; ALTER TABLE MC_SNIPPET_REF_SNIPPET DROP CONSTRAINT MCSNPPTRFSNIPPETSNPPTD; ALTER TABLE MC_SNIPPET_REF_TYPE DROP CONSTRAINT MCSNPPETREFTYPESNPPTID; ALTER TABLE MC_SNIPPET_TYPED_VAR DROP CONSTRAINT MCSNPPTTYPEDVARSNPPTID; ALTER TABLE MC_SNIPPET_REF_VAR DROP CONSTRAINT MCSNPPETREFVARSNPPETID; ALTER TABLE MC_OAUTH_APPLICATION DROP CONSTRAINT MCOAUTHAPPLICATIONP_ID; ALTER TABLE MC_OAUTH_APPLICATION DROP CONSTRAINT MC_OAUTH_APPLICATION0; ALTER TABLE MC_OAUTH_USER DROP CONSTRAINT MCAUTHUSERPPLICATIONID; ALTER TABLE MC_OAUTH_USER DROP CONSTRAINT

Database	Create Tables	Drop Tables
	NOT NULL, LOG_TO_DB BIT default 0 NOT NULL, LOG_TO_EMAIL BIT default 0 NOT NULL, LOG_TO_LOG4J BIT default 0 NOT NULL, MAX_CONCURRENT_ROBOTS INTEGER NULL, MAX_QUEUED_ROBOTS INTEGER NULL, PROF_ENABLED BIT default 0 NOT NULL, PROF_FILE_OUTPUT_APPEND BIT default 0 NOT NULL, PROF_FILE_OUTPUT_FILE NVARCHAR(255) NULL, PROF_LOG_PAGE_URL BIT default 0 NOT NULL, PROF_OUTPUT_TARGET NVARCHAR(20) NULL, PROF_TYPE NVARCHAR(20) NULL, PROF_THRESHOLD INTEGER NULL, EMAIL_LOG_TO_ADDRESS NVARCHAR(255) NULL, PRIMARY KEY (ID)); CREATE TABLE MC_DATABASE (ID NUMERIC(19) IDENTITY NOT NULL, HOST NVARCHAR(255) NOT NULL, MAX_ACTIVE_CONNECTIONS INTEGER NULL, MAX_IDLE_CONNECTIONS INTEGER NULL, NAME NVARCHAR(255) NOT NULL, PASSWORD NVARCHAR(255) NULL, SCHEMA_NAME NVARCHAR(255) NOT NULL, TYPE NVARCHAR(255) NOT NULL, USERNAME NVARCHAR(255) NULL, CLUSTER_SETTINGS_ID NUMERIC(19) NOT NULL, PRIMARY KEY (ID)); CREATE TABLE MC_PROXY_SERVER (ID NUMERIC(19) IDENTITY NOT NULL, EXCLUDED_HOST_NAMES NVARCHAR(1000) NULL, HOST_NAME NVARCHAR(255) NOT NULL, PASSWORD NVARCHAR(255) NULL, PORT_NUMBER INTEGER NULL, USERNAME NVARCHAR(255) NULL, CLUSTER_SETTINGS_ID NUMERIC(19) NOT NULL, PRIMARY KEY (ID)); CREATE TABLE MC_DATABASE_TYPE (ID NUMERIC(19) IDENTITY NOT NULL, DRIVER_CLASS NVARCHAR(255) NOT NULL, NAME NVARCHAR(255) NOT NULL, URL_TEMPLATE NVARCHAR(255) NOT NULL, VALIDATION_QUERY NVARCHAR(255) NOT NULL, PRIMARY KEY (ID)); CREATE TABLE MC_JAR_FILE (ID NUMERIC(19) IDENTITY NOT NULL, BYTES IMAGE NULL, CREATED DATETIME NULL, NAME NVARCHAR(255) NOT NULL, FILE_SIZE INTEGER NULL, PRIMARY KEY (ID)); CREATE TABLE APP_KAPPLET (ID NVARCHAR(40) NOT NULL, DTYPE NVARCHAR(31) NULL, CREATED_DATE DATETIME NOT NULL, DESCRIPTION TEXT NULL, NAME NVARCHAR(255) NOT NULL, OPTIONS TEXT NULL, P_ID NUMERIC(19) NOT NULL, ICON_ID NVARCHAR(255) NULL, PRIMARY KEY (ID)); CREATE TABLE APP_MASTER (ID NVARCHAR(40) NOT NULL, CREATEDBY NVARCHAR(255) NULL, ENABLED BIT default 0 NOT NULL, FLOW TEXT NULL, INITIAL_LABEL INTEGER NULL, KAPPLET_TYPE NVARCHAR(255) NOT NULL, LAST_INSTALLED DATETIME NULL, LAST_MODIFIED DATETIME NOT NULL, LAST_RUN DATETIME NULL, MODIFIEDBY NVARCHAR(255) NULL, PRIMARY KEY (ID)); CREATE TABLE APP_INSTALLED (ID NVARCHAR(40) NOT NULL, FAVORITE BIT default 0 NOT NULL, LAST_RUN DATETIME NULL, UPDATED_AT DATETIME NULL, PARENT	MC_OAUTH_USER_USER_ID; ALTER TABLE MC_OAUTH_USER DROP CONSTRAINT UNQ_MC_OAUTH_USER_0; ALTER TABLE MC_GROUP DROP CONSTRAINT UNQ_MC_GROUP_0; ALTER TABLE MC_USER_GROUP_REL DROP CONSTRAINT MCUSERGROUP_RELUSER_ID; ALTER TABLE MC_USER_GROUP_REL DROP CONSTRAINT MCUSERGROUPRELGROUP_ID; ALTER TABLE MC_USER_GROUP_REL DROP CONSTRAINT MC_USER_GROUP_REL0; ALTER TABLE MC_USER_PROP DROP CONSTRAINT MC_USER_PROP_USER_ID; ALTER TABLE MC_USER_PROP DROP CONSTRAINT UNQ_MC_USER_PROP_0; ALTER TABLE MC_TYPE_ATTR DROP CONSTRAINT MC_TYPE_ATTR_TYPE_ID; ALTER TABLE MC_DATABASE DROP CONSTRAINT MCDTBSECLSTRSETTINGSID; ALTER TABLE MC_PROXY_SERVER DROP CONSTRAINT MCPRXYSRVRCLSTRSTTNGSD; ALTER TABLE APP_KAPPLET DROP CONSTRAINT FK_APP_KAPPLET_ICON_ID; ALTER TABLE APP_KAPPLET DROP CONSTRAINT FK_APP_KAPPLET_P_ID; ALTER TABLE APP_MASTER DROP CONSTRAINT FK_APP_MASTER_ID; ALTER TABLE APP_INSTALLED DROP CONSTRAINT APP_INSTALLED_USER_ID; ALTER TABLE

Database	Create Tables	Drop Tables
	NVARCHAR(40) NOT NULL, USER_ID NVARCHAR(255) NOT NULL, PRIMARY KEY (ID)); CREATE TABLE APP_ICON (ID NVARCHAR(255) NOT NULL, ICON IMAGE NULL, filename NVARCHAR(255) NULL, mimeType NVARCHAR(255) NULL, PRIMARY KEY (ID)); CREATE TABLE APP_UPLOAD (ID NVARCHAR(255) NOT NULL, UPLOAD IMAGE NOT NULL, filename NVARCHAR(255) NOT NULL, mimeType NVARCHAR(255) NOT NULL, timestamp DATETIME NOT NULL, username NVARCHAR(255) NOT NULL, PRIMARY KEY (ID)); CREATE TABLE MC_KAPPLET_PROP (ID NUMERIC(19) IDENTITY NOT NULL, IS_NULL BIT default 0 NOT NULL, PROP_NAME NVARCHAR(255) NOT NULL, PROP_VALUE TEXT NULL, KAPPLET_ID NVARCHAR(40) NOT NULL, PRIMARY KEY (ID)); CREATE TABLE APP_SCHEDULE_ERROR (ID NUMERIC(19) IDENTITY NOT NULL, ERROR TEXT NULL, SCHEDULE_ID NUMERIC(19) NOT NULL, PRIMARY KEY (ID)); CREATE TABLE APP_SCHEDULE (ID NUMERIC(19) NOT NULL, DIRTY BIT default 0 NOT NULL, LASTRUN DATETIME NULL, NEXTRUN DATETIME NULL, PARAMETERS TEXT NULL, APP_ID NVARCHAR(40) NOT NULL, PRIMARY KEY (ID)); CREATE TABLE APP_SCHEDULE_VALUE (ID NVARCHAR(255) NOT NULL, VALUE IMAGE NOT NULL, SCHEDULE_ID NUMERIC(19) NOT NULL, PRIMARY KEY (ID)); CREATE TABLE MC_GLOBAL_KAPPLET_PROP (PROP_NAME NVARCHAR(255) NOT NULL, IS_NULL BIT default 0 NOT NULL, PROP_VALUE TEXT NULL, PRIMARY KEY (PROP_NAME)); CREATE TABLE APP_ACTION_EXECUTION (ID NUMERIC(19) NOT NULL, END_TIME DATETIME NULL, LABEL INTEGER NOT NULL, SEQ_ID INTEGER NULL, START_TIME DATETIME NULL, STATUS NVARCHAR(255) NULL, TASK_IDS NVARCHAR(255) NULL, VERSIONLOCK NUMERIC(19) NULL, KAPPLET_RUN_ID NUMERIC(19) NOT NULL, PRIMARY KEY (ID)); CREATE TABLE APP_DATA_ROW (ID NUMERIC(19) NOT NULL, TABLE_ID NUMERIC(19) NOT NULL, PRIMARY KEY (ID)); CREATE TABLE APP_DATA_TABLE (ID NUMERIC(19) NOT NULL, TABLE_NAME NVARCHAR(255) NOT NULL, FLOW_EXECUTION_ID NUMERIC(19) NOT NULL, PRIMARY KEY (ID)); CREATE TABLE APP_DATA_STR (ID NUMERIC(19) NOT NULL, NAME NVARCHAR(255) NULL, VALUE NVARCHAR(2000) NOT NULL, RESULT_ID NUMERIC(19) NOT NULL, PRIMARY KEY (ID)); CREATE TABLE APP_DATA_LOB (ID NUMERIC(19) NOT NULL, LARGE BIT default 0 NOT NULL, NAME NVARCHAR(255) NULL, VALUE TEXT NULL, RESULT_ID NUMERIC(19) NOT NULL, PRIMARY KEY (ID)); CREATE TABLE APP_FLOW_ERROR (ID NUMERIC(19) IDENTITY NOT NULL, ERROR TEXT NULL, EXECUTION_ID NUMERIC(19) NOT NULL, PRIMARY KEY (ID)); CREATE TABLE APP_RUN (ID NUMERIC(19) IDENTITY NOT	APP_INSTALLED DROP CONSTRAINT APP_INSTALLED_PARENT; ALTER TABLE APP_INSTALLED DROP CONSTRAINT FK_APP_INSTALLED_ID; ALTER TABLE MC_KAPPLET_PROP DROP CONSTRAINT MCKAPPLETPROPKAPPLETID; ALTER TABLE MC_KAPPLET_PROP DROP CONSTRAINT UNQ_MC_KAPPLET_PROP_0; ALTER TABLE APP_SCHEDULE_ERROR DROP CONSTRAINT PPSCHDULEERRORSCHDLEID; ALTER TABLE APP_SCHEDULE DROP CONSTRAINT FK_APP_SCHEDULE_APP_ID; ALTER TABLE APP_SCHEDULE_VALUE DROP CONSTRAINT PPSCHDULEVALUESCHDLEID; ALTER TABLE APP_ACTION_EXECUTION DROP CONSTRAINT PPCTNXECUTIONKPPLTRNID; ALTER TABLE APP_ACTION_EXECUTION DROP CONSTRAINT APP_ACTION_EXECUTION0; ALTER TABLE APP_DATA_ROW DROP CONSTRAINT APP_DATA_ROW_TABLE_ID; ALTER TABLE APP_DATA_TABLE DROP CONSTRAINT PPDTTABLEFLWXECUTIONID; ALTER TABLE APP_DATA_TABLE DROP CONSTRAINT UNQ_APP_DATA_TABLE_0; ALTER TABLE APP_DATA_STR DROP CONSTRAINT APP_DATA_STR_RESULT_ID; ALTER TABLE APP_DATA_LOB DROP CONSTRAINT APP_DATA_LOB_RESULT_ID; ALTER TABLE APP_FLOW_ERROR DROP CONSTRAINT

Database	Create Tables	Drop Tables
	NULL, ABORTED BIT default 0 NOT NULL, FLOW TEXT NULL, INITIAL_LABEL INTEGER NOT NULL, LASTEXECUTIONPATHCHANGE NUMERIC(19) NULL, OPTIONS TEXT NULL, PARAMETERS TEXT NULL, SIGNATURES TEXT NULL, VERSIONLOCK NUMERIC(19) NULL, KAPPLET_ID NVARCHAR(40) NOT NULL, PRIMARY KEY (ID)); CREATE TABLE APP_RUN_VALUE (ID NVARCHAR(255) NOT NULL, VALUE IMAGE NOT NULL, RUN_ID NUMERIC(19) NOT NULL, PRIMARY KEY (ID)); CREATE TABLE MC_FOLDER (ID NUMERIC(19) IDENTITY NOT NULL, DTYPE NVARCHAR(31) NULL, PATH NVARCHAR(255) NULL, PRIMARY KEY (ID)); CREATE TABLE MC_ROOT_FOLDER (ID NUMERIC(19) NOT NULL, P_ID NUMERIC(19) NOT NULL, PRIMARY KEY (ID)); CREATE TABLE MC_SUBFOLDER (ID NUMERIC(19) NOT NULL, NAME NVARCHAR(255) NOT NULL, PARENT_ID NUMERIC(19) NOT NULL, PRIMARY KEY (ID)); CREATE TABLE APP_RUN_APP_SCHEDULE (schedules_ID NUMERIC(19) NOT NULL, kappletRuns_ID NUMERIC(19) NOT NULL, PRIMARY KEY (schedules_ID, kappletRuns_ID)); ALTER TABLE MC_SCHEDULE ADD CONSTRAINT UNQ_MC_SCHEDULE_0 UNIQUE (P_ID, NAME); ALTER TABLE MC_BLOCK ADD CONSTRAINT UNQ_MC_BLOCK_0 UNIQUE (P_ID, NAME); ALTER TABLE MC_BLOCK_REFERENCE ADD CONSTRAINT MC_BLOCK_REFERENCE0 UNIQUE (PARENT, NAME); ALTER TABLE MC_ROBOT ADD CONSTRAINT UNQ_MC_ROBOT_0 UNIQUE (P_ID, NAME, FOLDER_EX); ALTER TABLE MC_TYPE ADD CONSTRAINT UNQ_MC_TYPE_0 UNIQUE (P_ID, NAME); ALTER TABLE MC_SNIPPET ADD CONSTRAINT UNQ_MC_SNIPPET_0 UNIQUE (P_ID, NAME); ALTER TABLE MC_RESOURCE_FILE ADD CONSTRAINT UNQ_MC_RESOURCE_FILE_0 UNIQUE (P_ID, NAME, FOLDER_EX); ALTER TABLE MC_GROUP_TO_ROLE_MAP ADD CONSTRAINT MC_GROUP_TO_ROLE_MAP0 UNIQUE (P_ID, ROLE_NAME, GROUP_NAME); ALTER TABLE MC_OAUTH_APPLICATION ADD CONSTRAINT MC_OAUTH_APPLICATION0 UNIQUE (P_ID, NAME); ALTER TABLE MC_OAUTH_USER ADD CONSTRAINT UNQ_MC_OAUTH_USER_0 UNIQUE (APPLICATION_ID, NAME); ALTER TABLE MC_GROUP ADD CONSTRAINT UNQ_MC_GROUP_0 UNIQUE (GROUP_NAME); ALTER TABLE MC_USER_GROUP_REL ADD CONSTRAINT MC_USER_GROUP_REL0 UNIQUE (USER_ID, GROUP_ID); ALTER TABLE MC_USER_PROP ADD CONSTRAINT UNQ_MC_USER_PROP_0 UNIQUE (ID, PROP_NAME); ALTER TABLE MC_KAPPLET_PROP ADD CONSTRAINT UNQ_MC_KAPPLET_PROP_0 UNIQUE (ID, PROP_NAME); ALTER TABLE APP_ACTION_EXECUTION ADD CONSTRAINT APP_ACTION_EXECUTION0 UNIQUE (KAPPLET_RUN_ID, LABEL); ALTER TABLE APP_DATA_TABLE ADD CONSTRAINT UNQ_APP_DATA_TABLE_0 UNIQUE (FLOW_EXECUTION_ID, TABLE_NAME); ALTER TABLE MC_SUBFOLDER ADD CONSTRAINT	PPFLOWERROREXECUTIONID; ALTER TABLE APP_RUN DROP CONSTRAINT FK_APP_RUN_KAPPLET_ID; ALTER TABLE APP_RUN_VALUE DROP CONSTRAINT APP_RUN_VALUE_RUN_ID; ALTER TABLE MC_ROOT_FOLDER DROP CONSTRAINT FK_MC_ROOT_FOLDER_P_ID; ALTER TABLE MC_ROOT_FOLDER DROP CONSTRAINT FK_MC_ROOT_FOLDER_ID; ALTER TABLE MC_SUBFOLDER DROP CONSTRAINT FK_MC_SUBFOLDER_ID; ALTER TABLE MC_SUBFOLDER DROP CONSTRAINT MC_SUBFOLDER_PARENT_ID; ALTER TABLE MC_SUBFOLDER DROP CONSTRAINT UNQ_MC_SUBFOLDER_0; ALTER TABLE APP_RUN_APP_SCHEDULE DROP CONSTRAINT PPRNPPSCHEDULEschdlsID; ALTER TABLE APP_RUN_APP_SCHEDULE DROP CONSTRAINT PPRNPPSCHDULEkppltRnsD; DROP TABLE MC_SCHEDULE; DROP TABLE MC_BLOCK; DROP TABLE MC_BLOCK_REFERENCE; DROP TABLE MC_BLOCK_JOB; DROP TABLE MC_ROBOT_BLOCK; DROP TABLE MC_MULTIPLE_ROBOT_BLOCK; DROP TABLE MC_CONTAINER_BLOCK; DROP TABLE MC_SEQUENTIAL_BLOCK; DROP TABLE MC_CONCURRENT_BLOCK; DROP TABLE MC_SCRIPT_BLOCK; DROP TABLE MC_FOLDER_ITEM;

Database	Create Tables	Drop Tables
	UNQ_MC_SUBFOLDER_0 UNIQUE (NAME, PARENT_ID); ALTER TABLE MC_SCHEDULE ADD CONSTRAINT MC_SCHEDULEBLOCKJOB_ID FOREIGN KEY (BLOCKJOB_ID) REFERENCES MC_BLOCK_JOB (ID); ALTER TABLE MC_SCHEDULE ADD CONSTRAINT FK_MC_SCHEDULE_P_ID FOREIGN KEY (P_ID) REFERENCES MC_PROJECT (ID); ALTER TABLE MC_SCHEDULE ADD CONSTRAINT MC_SCHEDULE_CLUSTER_ID FOREIGN KEY (CLUSTER_ID) REFERENCES MC_ROBOSERVER_CLUSTER (ID); ALTER TABLE MC_BLOCK ADD CONSTRAINT FK_MC_BLOCK_P_ID FOREIGN KEY (P_ID) REFERENCES MC_PROJECT (ID); ALTER TABLE MC_BLOCK_REFERENCE ADD CONSTRAINT MCBLOCK_REFERENCECHILD FOREIGN KEY (CHILD) REFERENCES MC_BLOCK (ID); ALTER TABLE MC_BLOCK_REFERENCE ADD CONSTRAINT MCBLOCKREFERENCEPARENT FOREIGN KEY (PARENT) REFERENCES MC_BLOCK (ID); ALTER TABLE MC_BLOCK_JOB ADD CONSTRAINT MC_BLOCK_JOB_BLOCK_ID FOREIGN KEY (BLOCK_ID) REFERENCES MC_BLOCK (ID); ALTER TABLE MC_ROBOT_BLOCK ADD CONSTRAINT FK_MC_ROBOT_BLOCK_ID FOREIGN KEY (ID) REFERENCES MC_BLOCK (ID); ALTER TABLE MC_ROBOT_BLOCK ADD CONSTRAINT MC_ROBOT_BLOCKROBOT_ID FOREIGN KEY (ROBOT_ID) REFERENCES MC_FOLDER_ITEM (ID); ALTER TABLE MC_MULTIPLE_ROBOT_BLOCK ADD CONSTRAINT MCMULTIPLEROBOTBLOCKID FOREIGN KEY (ID) REFERENCES MC_BLOCK (ID); ALTER TABLE MC_CONTAINER_BLOCK ADD CONSTRAINT MC_CONTAINER_BLOCK_ID FOREIGN KEY (ID) REFERENCES MC_BLOCK (ID); ALTER TABLE MC_SEQUENTIAL_BLOCK ADD CONSTRAINT MC_SEQUENTIAL_BLOCK_ID FOREIGN KEY (ID) REFERENCES MC_BLOCK (ID); ALTER TABLE MC_CONCURRENT_BLOCK ADD CONSTRAINT MC_CONCURRENT_BLOCK_ID FOREIGN KEY (ID) REFERENCES MC_BLOCK (ID); ALTER TABLE MC_SCRIPT_BLOCK ADD CONSTRAINT FK_MC_SCRIPT_BLOCK_ID FOREIGN KEY (ID) REFERENCES MC_BLOCK (ID); ALTER TABLE MC_FOLDER_ITEM ADD CONSTRAINT MC_FOLDER_ITEM_FOLDER FOREIGN KEY (FOLDER) REFERENCES MC_FOLDER (ID); ALTER TABLE MC_ROBOT ADD CONSTRAINT FK_MC_ROBOT_FOLDER_EX FOREIGN KEY (FOLDER_EX) REFERENCES MC_FOLDER (ID); ALTER TABLE MC_ROBOT ADD CONSTRAINT FK_MC_ROBOT_ID FOREIGN KEY (ID) REFERENCES MC_FOLDER_ITEM (ID); ALTER TABLE MC_ROBOT ADD CONSTRAINT FK_MC_ROBOT_P_ID FOREIGN KEY (P_ID) REFERENCES MC_PROJECT (ID); ALTER TABLE MC_TYPE ADD CONSTRAINT FK_MC_TYPE_P_ID FOREIGN KEY (P_ID) REFERENCES MC_PROJECT (ID); ALTER TABLE MC_TYPE ADD CONSTRAINT FK_MC_TYPE_ID FOREIGN	DROP TABLE MC_ROBOT; DROP TABLE MC_TYPE; DROP TABLE MC_SNIPPET; DROP TABLE MC_ROBOSERVER; DROP TABLE MC_SETTINGS; DROP TABLE MC_ROBOT_PARAMETER; DROP TABLE MC_ROBOT_REF_TYPE; DROP TABLE MC_ROBOT_REF_SNIPPET; DROP TABLE MC_ROBOT_TYPED_VARIABLE; DROP TABLE MC_ROBOT_REF_VAR; DROP TABLE MC_RESOURCE_FILE; DROP TABLE MC_PROJECT; DROP TABLE MC_ROBOSERVER_CLUSTER; DROP TABLE MC_GROUP_TO_ROLE_MAP; DROP TABLE MC_SNIPPET_PARAMETER; DROP TABLE MC_SNIPPET_REF_SNIPPET; DROP TABLE MC_SNIPPET_REF_TYPE; DROP TABLE MC_SNIPPET_TYPED_VAR; DROP TABLE MC_SNIPPET_REF_VAR; DROP TABLE MC_OAUTH_APPLICATION; DROP TABLE MC_OAUTH_USER; DROP TABLE MC_DSLICENSE; DROP TABLE MC_USER; DROP TABLE MC_GROUP; DROP TABLE MC_USER_GROUP_REL; DROP TABLE MC_USER_PROP; DROP TABLE MC_TYPE_ATTR; DROP TABLE MC_RS_CLUSTER_SETTINGS; DROP TABLE MC_DATABASE; DROP TABLE MC_PROXY_SERVER;

Database	Create Tables	Drop Tables
	KEY (ID) REFERENCES MC_FOLDER_ITEM (ID); ALTER TABLE MC_SNIPPET ADD CONSTRAINT FK_MC_SNIPPET_P_ID FOREIGN KEY (P_ID) REFERENCES MC_PROJECT (ID); ALTER TABLE MC_SNIPPET ADD CONSTRAINT FK_MC_SNIPPET_ID FOREIGN KEY (ID) REFERENCES MC_FOLDER_ITEM (ID); ALTER TABLE MC_ROBOSERVER ADD CONSTRAINT MCROBOSERVERCLUSTER_ID FOREIGN KEY (CLUSTER_ID) REFERENCES MC_ROBOSERVER_CLUSTER (ID); ALTER TABLE MC_ROBOT_PARAMETER ADD CONSTRAINT MCROBOTPARAMETERRBOTID FOREIGN KEY (ROBOT_ID) REFERENCES MC_FOLDER_ITEM (ID); ALTER TABLE MC_ROBOT_REF_TYPE ADD CONSTRAINT MCROBOTREFTYPEROBOT_ID FOREIGN KEY (ROBOT_ID) REFERENCES MC_FOLDER_ITEM (ID); ALTER TABLE MC_ROBOT_REF_SNIPPET ADD CONSTRAINT MCRBOTREFSNIPPETRBOTID FOREIGN KEY (ROBOT_ID) REFERENCES MC_FOLDER_ITEM (ID); ALTER TABLE MC_ROBOT_TYPED_VARIABLE ADD CONSTRAINT MCRBTTYPEDVARIABLERBTD FOREIGN KEY (ROBOT_ID) REFERENCES MC_FOLDER_ITEM (ID); ALTER TABLE MC_ROBOT_REF_VAR ADD CONSTRAINT MCROBOTREF_VARROBOT_ID FOREIGN KEY (ROBOT_ID) REFERENCES MC_FOLDER_ITEM (ID); ALTER TABLE MC_RESOURCE_FILE ADD CONSTRAINT MCRESOURCEFILEFOLDEREX FOREIGN KEY (FOLDER_EX) REFERENCES MC_FOLDER (ID); ALTER TABLE MC_RESOURCE_FILE ADD CONSTRAINT MC_RESOURCE_FILE_P_ID FOREIGN KEY (P_ID) REFERENCES MC_PROJECT (ID); ALTER TABLE MC_RESOURCE_FILE ADD CONSTRAINT FK_MC_RESOURCE_FILE_ID FOREIGN KEY (ID) REFERENCES MC_FOLDER_ITEM (ID); ALTER TABLE MC_PROJECT ADD CONSTRAINT MC_PROJECT_CLUSTER_ID FOREIGN KEY (CLUSTER_ID) REFERENCES MC_ROBOSERVER_CLUSTER (ID); ALTER TABLE MC_ROBOSERVER_CLUSTER ADD CONSTRAINT MCRBSRVRCLUSTERSTTNGSD FOREIGN KEY (SETTINGS_ID) REFERENCES MC_RS_CLUSTER_SETTINGS (ID); ALTER TABLE MC_GROUP_TO_ROLE_MAP ADD CONSTRAINT MCGROUPTO_ROLE_MAPP_ID FOREIGN KEY (P_ID) REFERENCES MC_PROJECT (ID); ALTER TABLE MC_SNIPPET_PARAMETER ADD CONSTRAINT MCSNPPTPARAMETERSNPPTD FOREIGN KEY (SNIPPET_ID) REFERENCES MC_FOLDER_ITEM (ID); ALTER TABLE MC_SNIPPET_REF_SNIPPET ADD CONSTRAINT MCSNPPTRFSNIPPETSNPPTD FOREIGN KEY (SNIPPET_ID) REFERENCES MC_FOLDER_ITEM (ID); ALTER TABLE MC_SNIPPET_REF_TYPE ADD CONSTRAINT MCSNPPETREFTYPESNPPTID FOREIGN KEY (SNIPPET_ID) REFERENCES MC_FOLDER_ITEM (ID); ALTER TABLE MC_SNIPPET_TYPED_VAR ADD CONSTRAINT MCSNPPTTYPEDVARSNPPTID FOREIGN KEY (SNIPPET_ID) REFERENCES MC_FOLDER_ITEM	DROP TABLE MC_DATABASE_TYPE; DROP TABLE MC_JAR_FILE; DROP TABLE APP_KAPPLET; DROP TABLE APP_MASTER; DROP TABLE APP_INSTALLED; DROP TABLE APP_ICON; DROP TABLE APP_UPLOAD; DROP TABLE MC_KAPPLET_PROP; DROP TABLE APP_SCHEDULE_ERROR; DROP TABLE APP_SCHEDULE; DROP TABLE APP_SCHEDULE_VALUE; DROP TABLE MC_GLOBAL_KAPPLET_PROP; DROP TABLE APP_ACTION_EXECUTION; DROP TABLE APP_DATA_ROW; DROP TABLE APP_DATA_TABLE; DROP TABLE APP_DATA_STR; DROP TABLE APP_DATA_LOB; DROP TABLE APP_FLOW_ERROR; DROP TABLE APP_RUN; DROP TABLE APP_RUN_VALUE; DROP TABLE MC_FOLDER; DROP TABLE MC_ROOT_FOLDER; DROP TABLE MC_SUBFOLDER; DROP TABLE APP_RUN_APP_SCHEDULE; DROP TABLE APP_SEQUENCES;

Database	Create Tables	Drop Tables
	(ID); ALTER TABLE MC_SNIPPET_REF_VAR ADD CONSTRAINT MCSNPPETREFVARSNPPETID FOREIGN KEY (SNIPPET_ID) REFERENCES MC_FOLDER_ITEM (ID); ALTER TABLE MC_OAUTH_APPLICATION ADD CONSTRAINT MCOAUTHAPPLICATIONP_ID FOREIGN KEY (P_ID) REFERENCES MC_PROJECT (ID); ALTER TABLE MC_OAUTH_USER ADD CONSTRAINT MCAUTHUSERPPLICATIONID FOREIGN KEY (APPLICATION_ID) REFERENCES MC_OAUTH_APPLICATION (ID); ALTER TABLE MC_OAUTH_USER ADD CONSTRAINT MC_OAUTH_USER_USER_ID FOREIGN KEY (USER_ID) REFERENCES MC_USER (USER_NAME); ALTER TABLE MC_USER_GROUP_REL ADD CONSTRAINT MCUSERGROUP_RELUSER_ID FOREIGN KEY (USER_ID) REFERENCES MC_USER (USER_NAME); ALTER TABLE MC_USER_GROUP_REL ADD CONSTRAINT MCUSERGROUPRELGROUP_ID FOREIGN KEY (GROUP_ID) REFERENCES MC_GROUP (ID); ALTER TABLE MC_USER_PROP ADD CONSTRAINT MC_USER_PROP_USER_ID FOREIGN KEY (USER_ID) REFERENCES MC_USER (USER_NAME); ALTER TABLE MC_TYPE_ATTR ADD CONSTRAINT MC_TYPE_ATTR_TYPE_ID FOREIGN KEY (TYPE_ID) REFERENCES MC_FOLDER_ITEM (ID); ALTER TABLE MC_DATABASE ADD CONSTRAINT MCDTBSECLSTRSETTINGSID FOREIGN KEY (CLUSTER_SETTINGS_ID) REFERENCES MC_RS_CLUSTER_SETTINGS (ID); ALTER TABLE MC_PROXY_SERVER ADD CONSTRAINT MCPRXYSRVRCLSTRSTTNGSD FOREIGN KEY (CLUSTER_SETTINGS_ID) REFERENCES MC_RS_CLUSTER_SETTINGS (ID); ALTER TABLE APP_KAPPLET ADD CONSTRAINT FK_APP_KAPPLET_ICON_ID FOREIGN KEY (ICON_ID) REFERENCES APP_ICON (ID); ALTER TABLE APP_KAPPLET ADD CONSTRAINT FK_APP_KAPPLET_P_ID FOREIGN KEY (P_ID) REFERENCES MC_PROJECT (ID); ALTER TABLE APP_MASTER ADD CONSTRAINT FK_APP_MASTER_ID FOREIGN KEY (ID) REFERENCES APP_KAPPLET (ID); ALTER TABLE APP_INSTALLED ADD CONSTRAINT APP_INSTALLED_USER_ID FOREIGN KEY (USER_ID) REFERENCES MC_USER (USER_NAME); ALTER TABLE APP_INSTALLED ADD CONSTRAINT APP_INSTALLED_PARENT FOREIGN KEY (PARENT) REFERENCES APP_KAPPLET (ID); ALTER TABLE APP_INSTALLED ADD CONSTRAINT FK_APP_INSTALLED_ID FOREIGN KEY (ID) REFERENCES APP_KAPPLET (ID); ALTER TABLE MC_KAPPLET_PROP ADD CONSTRAINT MCKAPPLETPROPKAPPLETID FOREIGN KEY (KAPPLET_ID) REFERENCES APP_KAPPLET (ID); ALTER TABLE APP_SCHEDULE_ERROR ADD CONSTRAINT PPSCHDULEERRORSCHDLEID FOREIGN KEY (SCHEDULE_ID) REFERENCES APP_SCHEDULE (ID); ALTER TABLE APP_SCHEDULE ADD CONSTRAINT FK_APP_SCHEDULE_APP_ID	

Database	Create Tables	Drop Tables
	FOREIGN KEY (APP_ID) REFERENCES APP_KAPPLET (ID); ALTER TABLE APP_SCHEDULE_VALUE ADD CONSTRAINT PPSCHDULEVALUESCHDLEID FOREIGN KEY (SCHEDULE_ID) REFERENCES APP_SCHEDULE (ID); ALTER TABLE APP_ACTION_EXECUTION ADD CONSTRAINT PPCTNXECUTIONKPPLTRNID FOREIGN KEY (KAPPLET_RUN_ID) REFERENCES APP_RUN (ID); ALTER TABLE APP_DATA_ROW ADD CONSTRAINT APP_DATA_ROW_TABLE_ID FOREIGN KEY (TABLE_ID) REFERENCES APP_DATA_TABLE (ID); ALTER TABLE APP_DATA_TABLE ADD CONSTRAINT PPDTTABLEFLWXECUTIONID FOREIGN KEY (FLOW_EXECUTION_ID) REFERENCES APP_ACTION_EXECUTION (ID); ALTER TABLE APP_DATA_STR ADD CONSTRAINT APP_DATA_STR_RESULT_ID FOREIGN KEY (RESULT_ID) REFERENCES APP_DATA_ROW (ID); ALTER TABLE APP_DATA_LOB ADD CONSTRAINT APP_DATA_LOB_RESULT_ID FOREIGN KEY (RESULT_ID) REFERENCES APP_DATA_ROW (ID); ALTER TABLE APP_FLOW_ERROR ADD CONSTRAINT PPFLOWERROREXECUTIONID FOREIGN KEY (EXECUTION_ID) REFERENCES APP_ACTION_EXECUTION (ID); ALTER TABLE APP_RUN ADD CONSTRAINT FK_APP_RUN_KAPPLET_ID FOREIGN KEY (KAPPLET_ID) REFERENCES APP_KAPPLET (ID); ALTER TABLE APP_RUN_VALUE ADD CONSTRAINT APP_RUN_VALUE_RUN_ID FOREIGN KEY (RUN_ID) REFERENCES APP_RUN (ID); ALTER TABLE MC_ROOT_FOLDER ADD CONSTRAINT FK_MC_ROOT_FOLDER_P_ID FOREIGN KEY (P_ID) REFERENCES MC_PROJECT (ID); ALTER TABLE MC_ROOT_FOLDER ADD CONSTRAINT FK_MC_ROOT_FOLDER_ID FOREIGN KEY (ID) REFERENCES MC_FOLDER (ID); ALTER TABLE MC_SUBFOLDER ADD CONSTRAINT FK_MC_SUBFOLDER_ID FOREIGN KEY (ID) REFERENCES MC_FOLDER (ID); ALTER TABLE MC_SUBFOLDER ADD CONSTRAINT MC_SUBFOLDER_PARENT_ID FOREIGN KEY (PARENT_ID) REFERENCES MC_FOLDER (ID); ALTER TABLE APP_RUN_APP_SCHEDULE ADD CONSTRAINT PPRNPPSCHEDULEschdlsID FOREIGN KEY (schedules_ID) REFERENCES APP_SCHEDULE (ID); ALTER TABLE APP_RUN_APP_SCHEDULE ADD CONSTRAINT PPRNPPSCHDULEkppltRnsD FOREIGN KEY (kappletRuns_ID) REFERENCES APP_RUN (ID); CREATE TABLE APP_SEQUENCES (TABLE_NAME NVARCHAR(50) NOT NULL, NEXT_ID NUMERIC(38) NULL, PRIMARY KEY (TABLE_NAME)); INSERT INTO APP_SEQUENCES(TABLE_NAME, NEXT_ID) values ('FLOW_ACTION_EXECUTION', 0); INSERT INTO APP_SEQUENCES(TABLE_NAME, NEXT_ID) values ('DATA_LOB', 0); INSERT INTO APP_SEQUENCES(TABLE_NAME, NEXT_ID) values ('DATA_TABLE', 0); INSERT INTO APP_SEQUENCES(TABLE_NAME, NEXT_ID) values ('DATA_STR', 0); INSERT INTO	

Database	Create Tables	Drop Tables
	APP_SEQUENCES(TABLE_NAME, NEXT_ID) values ('DATA_ROW', 0);	

SQL Scripts for Quartz Tables

Database	Create Tables	Drop Tables
IBM DB2	create table qrtz_job_details(job_name varchar(80) not null, job_group varchar(80) not null, description varchar(120), job_class_name varchar(128) not null, is_durable integer not null, is_volatile integer not null, is_stateful integer not null, requests_recovery integer not null, job_data blob(2000), primary key (job_name,job_group)); create table qrtz_job_listeners(job_name varchar(80) not null, job_group varchar(80) not null, job_listener varchar(80) not null, primary key (job_name,job_group,job_listener), foreign key (job_name,job_group) references qrtz_job_details(job_name,job_group)); create table qrtz_triggers(trigger_name varchar(80) not null, trigger_group varchar(80) not null, job_name varchar(80) not null, job_group varchar(80) not null, is_volatile integer not null, description varchar(120), next_fire_time bigint, prev_fire_time bigint, priority integer, trigger_state varchar(16) not null, trigger_type varchar(8) not null, start_time bigint not null, end_time bigint, calendar_name varchar(80), misfire_instr smallint, job_data blob(2000), primary key (trigger_name,trigger_group), foreign key (job_name,job_group) references qrtz_job_details (job_name,job_group)); create table qrtz_simple_triggers(trigger_name varchar(80) not null, trigger_group varchar(80) not null, repeat_count bigint not null, repeat_interval bigint not null, times_triggered bigint not null, primary key (trigger_name,trigger_group), foreign key (trigger_name,trigger_group) references qrtz_triggers (trigger_name,trigger_group));	drop table QRTZ_SIMPLE_TRIGGERS; drop table QRTZ_BLOB_TRIGGERS; drop table QRTZ_CRON_TRIGGERS; drop table QRTZ_TRIGGER_LISTENERS; drop table QRTZ_CALENDARS; drop table QRTZ_FIRED_TRIGGERS; drop table QRTZ_LOCKS; drop table QRTZ_PAUSED_TRIGGER_GRPS; drop table QRTZ_SCHEDULER_STATE; drop table QRTZ_JOB_LISTENERS; drop table QRTZ_TRIGGERS; drop table QRTZ_JOB_DETAILS;

Database	Create Tables	Drop Tables
	create table qrtz_cron_triggers(trigger_name varchar(80) not null, trigger_group varchar(80) not null, cron_expression varchar(120) not null, time_zone_id varchar(80), primary key (trigger_name,trigger_group), foreign key (trigger_name,trigger_group) references qrtz_triggers (trigger_name,trigger_group)); create table qrtz_blob_triggers(trigger_name varchar(80) not null, trigger_group varchar(80) not null, blob_data blob(2000), primary key (trigger_name,trigger_group), foreign key (trigger_name,trigger_group) references qrtz_triggers (trigger_name,trigger_group)); create table qrtz_trigger_listeners(trigger_name varchar(80) not null, trigger_group varchar(80) not null, trigger_listener varchar(80) not null, primary key (trigger_name,trigger_group,trigger_listener), foreign key (trigger_name,trigger_group) references qrtz_triggers (trigger_name,trigger_group)); create table qrtz_calendars(calendar_name varchar(80) not null, calendar blob(2000) not null, primary key (calendar_name)); create table qrtz_fired_triggers(entry_id varchar(95) not null, trigger_name varchar(80) not null, trigger_group varchar(80) not null, is_volatile integer not null, instance_name varchar(80) not null, fired_time bigint not null, priority integer not null, state varchar(16) not null, job_name varchar(80), job_group varchar(80), is_stateful integer, requests_recovery integer, primary key (entry_id)); create table qrtz_paused_trigger_grps(trigger_group varchar(80) not null, primary key (trigger_group)); create table qrtz_scheduler_state(instance_name varchar(80) not null, last_checkin_time bigint not null, checkin_interval bigint not null,	

Database	Create Tables	Drop Tables
	primary key (instance_name)); create table qrtz_locks(lock_name varchar(40) not null, primary key (lock_name)); insert into qrtz_locks values('TRIGGER_ACCESS'); insert into qrtz_locks values('JOB_ACCESS'); insert into qrtz_locks values('CALENDAR_ACCESS'); insert into qrtz_locks values('STATE_ACCESS'); insert into qrtz_locks values('MISFIRE_ACCESS');	
Derby	CREATE TABLE QRTZ_JOB_DETAILS (JOB_NAME VARCHAR(200) NOT NULL, JOB_GROUP VARCHAR(200) NOT NULL, DESCRIPTION VARCHAR(250) , JOB_CLASS_NAME VARCHAR(250) NOT NULL, IS_DURABLE VARCHAR(5) NOT NULL, IS_VOLATILE VARCHAR(5) NOT NULL, IS_STATEFUL VARCHAR(5) NOT NULL, REQUESTS_RECOVERY VARCHAR(5) NOT NULL, JOB_DATA BLOB, PRIMARY KEY (JOB_NAME,JOB_GROUP)); CREATE TABLE QRTZ_JOB_LISTENERS(JOB_NAME VARCHAR(200) NOT NULL, JOB_GROUP VARCHAR(200) NOT NULL, JOB_LISTENER VARCHAR(200) NOT NULL, PRIMARY KEY (JOB_NAME,JOB_GROUP,JOB_LISTENER), FOREIGN KEY (JOB_NAME,JOB_GROUP) REFERENCES QRTZ_JOB_DETAILS(JOB_NAME,JOB_GROUP)); CREATE TABLE QRTZ_TRIGGERS(TRIGGER_NAME VARCHAR(200) NOT NULL, TRIGGER_GROUP VARCHAR(200) NOT NULL, JOB_NAME VARCHAR(200) NOT NULL, JOB_GROUP VARCHAR(200) NOT NULL, IS_VOLATILE VARCHAR(5) NOT NULL, DESCRIPTION VARCHAR(250), NEXT_FIRE_TIME BIGINT, PREV_FIRE_TIME BIGINT, PRIORITY INTEGER, TRIGGER_STATE VARCHAR(16) NOT NULL, TRIGGER_TYPE VARCHAR(8) NOT NULL, START_TIME BIGINT NOT NULL, END_TIME BIGINT, CALENDAR_NAME VARCHAR(200), MISFIRE_INSTR SMALLINT, JOB_DATA BLOB, PRIMARY KEY (TRIGGER_NAME,TRIGGER_GROUP), FOREIGN KEY (JOB_NAME,JOB_GROUP) REFERENCES QRTZ_JOB_DETAILS(JOB_NAME,JOB_GROUP)); CREATE TABLE QRTZ_SIMPLE_TRIGGERS(TRIGGER_NAME VARCHAR(200) NOT NULL, TRIGGER_GROUP VARCHAR(200) NOT NULL, REPEAT_COUNT BIGINT NOT NULL, REPEAT_INTERVAL BIGINT NOT NULL, TIMES_TRIGGERED BIGINT NOT NULL, PRIMARY KEY	drop table QRTZ_SIMPLE_TRIGGERS; drop table QRTZ_BLOB_TRIGGERS; drop table QRTZ_CRON_TRIGGERS; drop table QRTZ_TRIGGER_LISTENERS; drop table QRTZ_CALENDARS; drop table QRTZ_FIRED_TRIGGERS; drop table QRTZ_LOCKS; drop table QRTZ_PAUSED_TRIGGER_GRPS; drop table QRTZ_SCHEDULER_STATE; drop table QRTZ_JOB_LISTENERS; drop table QRTZ_TRIGGERS; drop table QRTZ_JOB_DETAILS;

Database	Create Tables	Drop Tables
	(TRIGGER_NAME,TRIGGER_GROUP), FOREIGN KEY (TRIGGER_NAME,TRIGGER_GROUP) REFERENCES QRTZ_TRIGGERS (TRIGGER_NAME,TRIGGER_GROUP)); CREATE TABLE QRTZ_CRON_TRIGGERS(TRIGGER_NAME VARCHAR(200) NOT NULL, TRIGGER_GROUP VARCHAR(200) NOT NULL, CRON_EXPRESSION VARCHAR(120) NOT NULL, TIME_ZONE_ID VARCHAR(80), PRIMARY KEY (TRIGGER_NAME,TRIGGER_GROUP), FOREIGN KEY (TRIGGER_NAME,TRIGGER_GROUP) REFERENCES QRTZ_TRIGGERS (TRIGGER_NAME,TRIGGER_GROUP)); CREATE TABLE QRTZ_BLOB_TRIGGERS(TRIGGER_NAME VARCHAR(200) NOT NULL, TRIGGER_GROUP VARCHAR(200) NOT NULL, BLOB_DATA BLOB, PRIMARY KEY (TRIGGER_NAME,TRIGGER_GROUP), FOREIGN KEY (TRIGGER_NAME,TRIGGER_GROUP) REFERENCES QRTZ_TRIGGERS (TRIGGER_NAME,TRIGGER_GROUP)); CREATE TABLE QRTZ_TRIGGER_LISTENERS(TRIGGER_NAME VARCHAR(200) NOT NULL, TRIGGER_GROUP VARCHAR(200) NOT NULL, TRIGGER_LISTENER VARCHAR(200) NOT NULL, PRIMARY KEY (TRIGGER_NAME,TRIGGER_GROUP,TRIGGER_LISTENER), FOREIGN KEY (TRIGGER_NAME,TRIGGER_GROUP) REFERENCES QRTZ_TRIGGERS (TRIGGER_NAME,TRIGGER_GROUP)); CREATE TABLE QRTZ_CALENDARS(CALENDAR_NAME VARCHAR(200) NOT NULL, CALENDAR BLOB NOT NULL, PRIMARY KEY (CALENDAR_NAME)); CREATE TABLE QRTZ_PAUSED_TRIGGER_GRPS (TRIGGER_GROUP VARCHAR(200) NOT NULL, PRIMARY KEY (TRIGGER_GROUP)); CREATE TABLE QRTZ_FIRED_TRIGGERS(ENTRY_ID VARCHAR(95) NOT NULL, TRIGGER_NAME VARCHAR(200) NOT NULL, TRIGGER_GROUP VARCHAR(200) NOT NULL, IS_VOLATILE VARCHAR(5) NOT NULL, INSTANCE_NAME VARCHAR(200) NOT NULL, FIRED_TIME BIGINT NOT NULL, PRIORITY INTEGER NOT NULL, STATE VARCHAR(16) NOT NULL, JOB_NAME VARCHAR(200), JOB_GROUP VARCHAR(200), IS_STATEFUL VARCHAR(5),	

Database	Create Tables	Drop Tables
	REQUESTS_RECOVERY VARCHAR(5), PRIMARY KEY (ENTRY_ID)); CREATE TABLE QRTZ_SCHEDULER_STATE (INSTANCE_NAME VARCHAR(200) NOT NULL, LAST_CHECKIN_TIME BIGINT NOT NULL, CHECKIN_INTERVAL BIGINT NOT NULL, PRIMARY KEY (INSTANCE_NAME)); CREATE TABLE QRTZ_LOCKS (LOCK_NAME VARCHAR(40) NOT NULL, PRIMARY KEY (LOCK_NAME)); INSERT INTO QRTZ_LOCKS VALUES('TRIGGER_ACCESS'); INSERT INTO QRTZ_LOCKS VALUES('JOB_ACCESS'); INSERT INTO QRTZ_LOCKS VALUES('CALENDAR_ACCESS'); INSERT INTO QRTZ_LOCKS VALUES('STATE_ACCESS'); INSERT INTO QRTZ_LOCKS VALUES('MISFIRE_ACCESS');	
MySQL	CREATE TABLE QRTZ_JOB_DETAILS (JOB_NAME VARCHAR(80) NOT NULL, JOB_GROUP VARCHAR(80) NOT NULL, DESCRIPTION VARCHAR(80) NULL, JOB_CLASS_NAME VARCHAR(150) NOT NULL, IS_DURABLE VARCHAR(1) NOT NULL, IS_VOLATILE VARCHAR(1) NOT NULL, IS_STATEFUL VARCHAR(1) NOT NULL, REQUESTS_RECOVERY VARCHAR(1) NOT NULL, JOB_DATA BLOB NULL, PRIMARY KEY (JOB_NAME,JOB_GROUP)); CREATE TABLE QRTZ_JOB_LISTENERS (JOB_NAME VARCHAR(80) NOT NULL, JOB_GROUP VARCHAR(80) NOT NULL, JOB_LISTENER VARCHAR(100) NOT NULL, PRIMARY KEY (JOB_NAME,JOB_GROUP,JOB_LISTENER), FOREIGN KEY (JOB_NAME,JOB_GROUP) REFERENCES QRTZ_JOB_DETAILS (JOB_NAME,JOB_GROUP)); CREATE TABLE QRTZ_TRIGGERS (TRIGGER_NAME VARCHAR(80) NOT NULL, TRIGGER_GROUP VARCHAR(80) NOT NULL, JOB_NAME VARCHAR(80) NOT NULL, JOB_GROUP VARCHAR(80) NOT NULL, IS_VOLATILE VARCHAR(1) NOT NULL, DESCRIPTION VARCHAR(100) NULL, NEXT_FIRE_TIME BIGINT(13) NULL, PREV_FIRE_TIME BIGINT(13) NULL, PRIORITY INTEGER NULL, TRIGGER_STATE VARCHAR(16) NOT NULL, TRIGGER_TYPE VARCHAR(8) NOT NULL, START_TIME BIGINT(13) NOT NULL, END_TIME BIGINT(13) NULL, CALENDAR_NAME VARCHAR(200) NULL, MISFIRE_INSTR SMALLINT(2) NULL, JOB_DATA BLOB NULL, PRIMARY KEY (TRIGGER_NAME,TRIGGER_GROUP), FOREIGN KEY (JOB_NAME,JOB_GROUP) REFERENCES QRTZ_JOB_DETAILS	drop table if exists QRTZ_SIMPLE_TRIGGERS; drop table if exists QRTZ_BLOB_TRIGGERS; drop table if exists QRTZ_CRON_TRIGGERS; drop table if exists QRTZ_TRIGGER_LISTENERS; drop table if exists QRTZ_CALENDARS; drop table if exists QRTZ_FIRED_TRIGGERS; drop table if exists QRTZ_LOCKS; drop table if exists QRTZ_PAUSED_TRIGGER_GRPS; drop table if exists QRTZ_SCHEDULER_STATE; drop table if exists QRTZ_JOB_LISTENERS; drop table if exists QRTZ_TRIGGERS; drop table if exists QRTZ_JOB_DETAILS;

Database	Create Tables	Drop Tables
	(JOB_NAME,JOB_GROUP)); CREATE TABLE QRTZ_SIMPLE_TRIGGERS (TRIGGER_NAME VARCHAR(80) NOT NULL, TRIGGER_GROUP VARCHAR(80) NOT NULL, REPEAT_COUNT BIGINT(7) NOT NULL, REPEAT_INTERVAL BIGINT(12) NOT NULL, TIMES_TRIGGERED BIGINT(10) NOT NULL, PRIMARY KEY (TRIGGER_NAME,TRIGGER_GROUP), FOREIGN KEY (TRIGGER_NAME,TRIGGER_GROUP) REFERENCES QRTZ_TRIGGERS (TRIGGER_NAME,TRIGGER_GROUP)); CREATE TABLE QRTZ_CRON_TRIGGERS (TRIGGER_NAME VARCHAR(80) NOT NULL, TRIGGER_GROUP VARCHAR(80) NOT NULL, CRON_EXPRESSION VARCHAR(200) NOT NULL, TIME_ZONE_ID VARCHAR(80), PRIMARY KEY (TRIGGER_NAME,TRIGGER_GROUP), FOREIGN KEY (TRIGGER_NAME,TRIGGER_GROUP) REFERENCES QRTZ_TRIGGERS (TRIGGER_NAME,TRIGGER_GROUP)); CREATE TABLE QRTZ_BLOB_TRIGGERS (TRIGGER_NAME VARCHAR(80) NOT NULL, TRIGGER_GROUP VARCHAR(80) NOT NULL, BLOB_DATA BLOB NULL, PRIMARY KEY (TRIGGER_NAME,TRIGGER_GROUP), FOREIGN KEY (TRIGGER_NAME,TRIGGER_GROUP) REFERENCES QRTZ_TRIGGERS (TRIGGER_NAME,TRIGGER_GROUP)); CREATE TABLE QRTZ_TRIGGER_LISTENERS (TRIGGER_NAME VARCHAR(80) NOT NULL, TRIGGER_GROUP VARCHAR(80) NOT NULL, TRIGGER_LISTENER VARCHAR(80) NOT NULL, PRIMARY KEY (TRIGGER_NAME,TRIGGER_GROUP,TRIGGER_LISTENER), FOREIGN KEY (TRIGGER_NAME,TRIGGER_GROUP) REFERENCES QRTZ_TRIGGERS (TRIGGER_NAME,TRIGGER_GROUP)); CREATE TABLE QRTZ_CALENDARS (CALENDAR_NAME VARCHAR(200) NOT NULL, CALENDAR BLOB NOT NULL, PRIMARY KEY (CALENDAR_NAME)); CREATE TABLE QRTZ_PAUSED_TRIGGER_GRPS (TRIGGER_GROUP VARCHAR(200) NOT NULL, PRIMARY KEY (TRIGGER_GROUP)); CREATE TABLE QRTZ_FIRED_TRIGGERS (ENTRY_ID VARCHAR(95) NOT NULL, TRIGGER_NAME VARCHAR(80) NOT NULL, TRIGGER_GROUP VARCHAR(80) NOT NULL, IS_VOLATILE VARCHAR(1) NOT NULL, INSTANCE_NAME VARCHAR(200) NOT NULL, FIRED_TIME BIGINT(13) NOT NULL, PRIORITY INTEGER NOT	

Database	Create Tables	Drop Tables
	NULL, STATE VARCHAR(16) NOT NULL, JOB_NAME VARCHAR(200) NULL, JOB_GROUP VARCHAR(200) NULL, IS_STATEFUL VARCHAR(1) NULL, REQUESTS_RECOVERY VARCHAR(1) NULL, PRIMARY KEY (ENTRY_ID)); CREATE TABLE QRTZ_SCHEDULER_STATE (INSTANCE_NAME VARCHAR(200) NOT NULL, LAST_CHECKIN_TIME BIGINT(13) NOT NULL, CHECKIN_INTERVAL BIGINT(13) NOT NULL, PRIMARY KEY (INSTANCE_NAME)); CREATE TABLE QRTZ_LOCKS (LOCK_NAME VARCHAR(40) NOT NULL, PRIMARY KEY (LOCK_NAME)); INSERT INTO QRTZ_LOCKS values('TRIGGER_ACCESS'); INSERT INTO QRTZ_LOCKS values('JOB_ACCESS'); INSERT INTO QRTZ_LOCKS values('CALENDAR_ACCESS'); INSERT INTO QRTZ_LOCKS values('STATE_ACCESS'); INSERT INTO QRTZ_LOCKS values('MISFIRE_ACCESS');	
Oracle	CREATE TABLE QRTZ_JOB_DETAILS (JOB_NAME VARCHAR2(200) NOT NULL, JOB_GROUP VARCHAR2(200) NOT NULL, DESCRIPTION VARCHAR2(250) NULL, JOB_CLASS_NAME VARCHAR2(250) NOT NULL, IS_DURABLE VARCHAR2(1) NOT NULL, IS_VOLATILE VARCHAR2(1) NOT NULL, IS_STATEFUL VARCHAR2(1) NOT NULL, REQUESTS_RECOVERY VARCHAR2(1) NOT NULL, JOB_DATA BLOB NULL, PRIMARY KEY (JOB_NAME,JOB_GROUP)); CREATE TABLE QRTZ_JOB_LISTENERS (JOB_NAME VARCHAR2(200) NOT NULL, JOB_GROUP VARCHAR2(200) NOT NULL, JOB_LISTENER VARCHAR2(200) NOT NULL, PRIMARY KEY (JOB_NAME,JOB_GROUP,JOB_LISTENER), FOREIGN KEY (JOB_NAME,JOB_GROUP) REFERENCES QRTZ_JOB_DETAILS(JOB_NAME,JOB_GROUP)); CREATE TABLE QRTZ_TRIGGERS (TRIGGER_NAME VARCHAR2(200) NOT NULL, TRIGGER_GROUP VARCHAR2(200) NOT NULL, JOB_NAME VARCHAR2(200) NOT NULL, JOB_GROUP VARCHAR2(200) NOT NULL, IS_VOLATILE VARCHAR2(1) NOT NULL, DESCRIPTION VARCHAR2(250) NULL, NEXT_FIRE_TIME NUMBER(13) NULL, PREV_FIRE_TIME NUMBER(13) NULL, PRIORITY NUMBER(13) NULL, TRIGGER_STATE VARCHAR2(16) NOT NULL, TRIGGER_TYPE VARCHAR2(8) NOT NULL, START_TIME NUMBER(13) NOT NULL, END_TIME NUMBER(13) NULL,	drop table QRTZ_SIMPLE_TRIGGERS; drop table QRTZ_BLOB_TRIGGERS; drop table QRTZ_CRON_TRIGGERS; drop table QRTZ_TRIGGER_LISTENERS; drop table QRTZ_CALENDARS; drop table QRTZ_FIRED_TRIGGERS; drop table QRTZ_LOCKS; drop table QRTZ_PAUSED_TRIGGER_GRPS; drop table QRTZ_SCHEDULER_STATE; drop table QRTZ_JOB_LISTENERS; drop table QRTZ_TRIGGERS; drop table QRTZ_JOB_DETAILS;

Database	Create Tables	Drop Tables
	CALENDAR_NAME VARCHAR2(200) NULL, MISFIRE_INSTR NUMBER(2) NULL, JOB_DATA BLOB NULL, PRIMARY KEY (TRIGGER_NAME,TRIGGER_GROUP), FOREIGN KEY (JOB_NAME,JOB_GROUP) REFERENCES QRTZ_JOB_DETAILS(JOB_NAME,JOB_GROUP)); CREATE TABLE QRTZ_SIMPLE_TRIGGERS (TRIGGER_NAME VARCHAR2(200) NOT NULL, TRIGGER_GROUP VARCHAR2(200) NOT NULL, REPEAT_COUNT NUMBER(7) NOT NULL, REPEAT_INTERVAL NUMBER(12) NOT NULL, TIMES_TRIGGERED NUMBER(10) NOT NULL, PRIMARY KEY (TRIGGER_NAME,TRIGGER_GROUP), FOREIGN KEY (TRIGGER_NAME,TRIGGER_GROUP) REFERENCES QRTZ_TRIGGERS (TRIGGER_NAME,TRIGGER_GROUP)); CREATE TABLE QRTZ_CRON_TRIGGERS (TRIGGER_NAME VARCHAR2(200) NOT NULL, TRIGGER_GROUP VARCHAR2(200) NOT NULL, CRON_EXPRESSION VARCHAR2(120) NOT NULL, TIME_ZONE_ID VARCHAR2(80), PRIMARY KEY (TRIGGER_NAME,TRIGGER_GROUP), FOREIGN KEY (TRIGGER_NAME,TRIGGER_GROUP) REFERENCES QRTZ_TRIGGERS (TRIGGER_NAME,TRIGGER_GROUP)); CREATE TABLE QRTZ_BLOB_TRIGGERS (TRIGGER_NAME VARCHAR2(200) NOT NULL, TRIGGER_GROUP VARCHAR2(200) NOT NULL, BLOB_DATA BLOB NULL, PRIMARY KEY (TRIGGER_NAME,TRIGGER_GROUP), FOREIGN KEY (TRIGGER_NAME,TRIGGER_GROUP) REFERENCES QRTZ_TRIGGERS (TRIGGER_NAME,TRIGGER_GROUP)); CREATE TABLE QRTZ_TRIGGER_LISTENERS (TRIGGER_NAME VARCHAR2(200) NOT NULL, TRIGGER_GROUP VARCHAR2(200) NOT NULL, TRIGGER_LISTENER VARCHAR2(200) NOT NULL, PRIMARY KEY (TRIGGER_NAME,TRIGGER_GROUP,TRIGGER_LISTENER), FOREIGN KEY (TRIGGER_NAME,TRIGGER_GROUP) REFERENCES QRTZ_TRIGGERS (TRIGGER_NAME,TRIGGER_GROUP)); CREATE TABLE QRTZ_CALENDARS (CALENDAR_NAME VARCHAR2(200) NOT NULL, CALENDAR BLOB NOT NULL, PRIMARY KEY (CALENDAR_NAME)); CREATE TABLE QRTZ_PAUSED_TRIGGER_GRPS (TRIGGER_GROUP VARCHAR2(200) NOT NULL, PRIMARY KEY (TRIGGER_GROUP));	

Database	Create Tables	Drop Tables

	CREATE TABLE QRTZ_FIRED_TRIGGERS
	(ENTRY_ID VARCHAR2(95) NOT NULL,
	TRIGGER_NAME VARCHAR2(200) NOT NULL,
	TRIGGER_GROUP VARCHAR2(200) NOT
	NULL, IS_VOLATILE VARCHAR2(1) NOT
	NULL, INSTANCE_NAME VARCHAR2(200)
	NOT NULL, FIRED_TIME NUMBER(13)
	NOT NULL, PRIORITY NUMBER(13)
	NOT NULL, STATE VARCHAR2(16) NOT
	NULL, JOB_NAME VARCHAR2(200) NULL,
	JOB_GROUP VARCHAR2(200) NULL,
	IS_STATEFUL VARCHAR2(1) NULL,
	REQUESTS_RECOVERY VARCHAR2(1)
	NULL, PRIMARY KEY (ENTRY_ID));
	CREATE TABLE QRTZ_SCHEDULER_STATE
	(INSTANCE_NAME VARCHAR2(200) NOT
	NULL, LAST_CHECKIN_TIME NUMBER(13)
	NOT NULL, CHECKIN_INTERVAL
	NUMBER(13) NOT NULL, PRIMARY
	KEY (INSTANCE_NAME)); CREATE
	TABLE QRTZ_LOCKS (LOCK_NAME
	VARCHAR2(40) NOT NULL, PRIMARY
	KEY (LOCK_NAME)); INSERT INTO
	QRTZ_LOCKS values('TRIGGER_ACCESS');
	INSERT INTO QRTZ_LOCKS
	values('JOB_ACCESS');
	INSERT INTO QRTZ_LOCKS
	values('CALENDAR_ACCESS');
	INSERT INTO QRTZ_LOCKS
	values('STATE_ACCESS');
	INSERT INTO QRTZ_LOCKS
	values('MISFIRE_ACCESS'); create
	index IDX_QRTZ_J_REQ_RECOVERY on
	QRTZ_JOB_DETAILS(REQUESTS_RECOVERY);
	create index
	IDX_QRTZ_T_NEXT_FIRE_TIME on
	QRTZ_TRIGGERS(NEXT_FIRE_TIME);
	create index IDX_QRTZ_T_STATE
	on QRTZ_TRIGGERS(TRIGGER_STATE);
	create index IDX_QRTZ_T_NFT_ST on
	QRTZ_TRIGGERS(NEXT_FIRE_TIME,TRIGGER_STATE);
	create index IDX_QRTZ_T_VOLATILE on
	QRTZ_TRIGGERS(IS_VOLATILE); create
	index IDX_QRTZ_FT_TRIG_NAME on
	QRTZ_FIRED_TRIGGERS(TRIGGER_NAME);
	create index
	IDX_QRTZ_FT_TRIG_GROUP on
	QRTZ_FIRED_TRIGGERS(TRIGGER_GROUP);
	create index
	IDX_QRTZ_FT_TRIG_NM_GP on
	QRTZ_FIRED_TRIGGERS(TRIGGER_NAME,TRIGGER_GROUP);
	create index
	IDX_QRTZ_FT_TRIG_VOLATILE on
	QRTZ_FIRED_TRIGGERS(IS_VOLATILE);
	create index
	IDX_QRTZ_FT_TRIG_INST_NAME on
	QRTZ_FIRED_TRIGGERS(INSTANCE_NAME);
	create index IDX_QRTZ_FT_JOB_NAME	

Database	Create Tables	Drop Tables

	on QRTZ_FIRED_TRIGGERS(JOB_NAME);
	create index IDX_QRTZ_FT_JOB_GROUP
	on QRTZ_FIRED_TRIGGERS(JOB_GROUP);
	create index
	IDX_QRTZ_FT_JOB_STATEFUL on
	QRTZ_FIRED_TRIGGERS(IS_STATEFUL);
	create index
	IDX_QRTZ_FT_JOB_REQ_RECOVERY on
	QRTZ_FIRED_TRIGGERS(REQUESTS_RECOVERY);
	commit;	

Microsoft SQL Server	CREATE TABLE QRTZ_JOB_DETAILS (JOB_NAME VARCHAR(80) NOT NULL, JOB_GROUP VARCHAR(80) NOT NULL, DESCRIPTION VARCHAR(80) NULL, JOB_CLASS_NAME VARCHAR(150) NOT NULL, IS_DURABLE BOOLEAN NOT NULL, IS_VOLATILE BOOLEAN NOT NULL, IS_STATEFUL BOOLEAN NOT NULL, REQUESTS_RECOVERY BOOLEAN NOT NULL, JOB_DATA BYTEA NULL, PRIMARY KEY (JOB_NAME,JOB_GROUP)); CREATE TABLE QRTZ_JOB_LISTENERS (JOB_NAME VARCHAR(80) NOT NULL, JOB_GROUP VARCHAR(80) NOT NULL, JOB_LISTENER VARCHAR(100) NOT NULL, PRIMARY KEY (JOB_NAME,JOB_GROUP,JOB_LISTENER), FOREIGN KEY (JOB_NAME,JOB_GROUP) REFERENCES QRTZ_JOB_DETAILS(JOB_NAME,JOB_GROUP)); CREATE TABLE QRTZ_TRIGGERS (TRIGGER_NAME VARCHAR(80) NOT NULL, TRIGGER_GROUP VARCHAR(80) NOT NULL, JOB_NAME VARCHAR(80) NOT NULL, JOB_GROUP VARCHAR(80) NOT NULL, IS_VOLATILE BOOLEAN NOT NULL, DESCRIPTION VARCHAR(100) NULL, NEXT_FIRE_TIME BIGINT NULL, PREV_FIRE_TIME BIGINT NULL, PRIORITY INTEGER NULL, TRIGGER_STATE VARCHAR(16) NOT NULL, TRIGGER_TYPE VARCHAR(8) NOT NULL, START_TIME BIGINT NOT NULL, END_TIME BIGINT NULL, CALENDAR_NAME VARCHAR(200) NULL, MISFIRE_INSTR SMALLINT NULL, JOB_DATA BYTEA NULL, PRIMARY KEY (TRIGGER_NAME,TRIGGER_GROUP), FOREIGN KEY (JOB_NAME,JOB_GROUP) REFERENCES QRTZ_JOB_DETAILS(JOB_NAME,JOB_GROUP)); CREATE TABLE QRTZ_SIMPLE_TRIGGERS (TRIGGER_NAME VARCHAR(80) NOT NULL, TRIGGER_GROUP VARCHAR(80) NOT NULL, REPEAT_COUNT BIGINT NOT NULL, REPEAT_INTERVAL BIGINT NOT NULL, TIMES_TRIGGERED BIGINT NOT NULL, PRIMARY KEY (TRIGGER_NAME,TRIGGER_GROUP),	drop table QRTZ_SIMPLE_TRIGGERS; drop table QRTZ_BLOB_TRIGGERS; drop table QRTZ_CRON_TRIGGERS; drop table QRTZ_TRIGGER_LISTENERS; drop table QRTZ_CALENDARS; drop table QRTZ_FIRED_TRIGGERS; drop table QRTZ_LOCKS; drop table QRTZ_PAUSED_TRIGGER_GRPS; drop table QRTZ_SCHEDULER_STATE; drop table QRTZ_JOB_LISTENERS; drop table QRTZ_TRIGGERS; drop table QRTZ_JOB_DETAILS;

Database	Create Tables	Drop Tables
	FOREIGN KEY (TRIGGER_NAME,TRIGGER_GROUP) REFERENCES QRTZ_TRIGGERS(TRIGGER_NAME,TRIGGER_GROUP)); CREATE TABLE QRTZ_CRON_TRIGGERS (TRIGGER_NAME VARCHAR(80) NOT NULL, TRIGGER_GROUP VARCHAR(80) NOT NULL, CRON_EXPRESSION VARCHAR(200) NOT NULL, TIME_ZONE_ID VARCHAR(80), PRIMARY KEY (TRIGGER_NAME,TRIGGER_GROUP), FOREIGN KEY (TRIGGER_NAME,TRIGGER_GROUP) REFERENCES QRTZ_TRIGGERS(TRIGGER_NAME,TRIGGER_GROUP)); CREATE TABLE QRTZ_BLOB_TRIGGERS (TRIGGER_NAME VARCHAR(80) NOT NULL, TRIGGER_GROUP VARCHAR(80) NOT NULL, BLOB_DATA BYTEA NULL, PRIMARY KEY (TRIGGER_NAME,TRIGGER_GROUP), FOREIGN KEY (TRIGGER_NAME,TRIGGER_GROUP) REFERENCES QRTZ_TRIGGERS(TRIGGER_NAME,TRIGGER_GROUP)); CREATE TABLE QRTZ_TRIGGER_LISTENERS (TRIGGER_NAME VARCHAR(80) NOT NULL, TRIGGER_GROUP VARCHAR(80) NOT NULL, TRIGGER_LISTENER VARCHAR(80) NOT NULL, PRIMARY KEY (TRIGGER_NAME,TRIGGER_GROUP,TRIGGER_LISTENER), FOREIGN KEY (TRIGGER_NAME,TRIGGER_GROUP) REFERENCES QRTZ_TRIGGERS(TRIGGER_NAME,TRIGGER_GROUP)); CREATE TABLE QRTZ_CALENDARS (CALENDAR_NAME VARCHAR(200) NOT NULL, CALENDAR BYTEA NOT NULL, PRIMARY KEY (CALENDAR_NAME)); CREATE TABLE QRTZ_PAUSED_TRIGGER_GRPS (TRIGGER_GROUP VARCHAR(200) NOT NULL, PRIMARY KEY (TRIGGER_GROUP)); CREATE TABLE QRTZ_FIRED_TRIGGERS (ENTRY_ID VARCHAR(95) NOT NULL, TRIGGER_NAME VARCHAR(80) NOT NULL, TRIGGER_GROUP VARCHAR(80) NOT NULL, IS_VOLATILE BOOLEAN NOT NULL, INSTANCE_NAME VARCHAR(200) NOT NULL, FIRED_TIME BIGINT NOT NULL, PRIORITY INTEGER NOT NULL, STATE VARCHAR(16) NOT NULL, JOB_NAME VARCHAR(200) NULL, JOB_GROUP VARCHAR(200) NULL, IS_STATEFUL BOOLEAN NULL, REQUESTS_RECOVERY BOOLEAN NULL, PRIMARY KEY (ENTRY_ID)); CREATE TABLE QRTZ_SCHEDULER_STATE (INSTANCE_NAME VARCHAR(200) NOT NULL, LAST_CHECKIN_TIME BIGINT NOT	

Database	Create Tables	Drop Tables
	NULL, CHECKIN_INTERVAL BIGINT NOT NULL, PRIMARY KEY (INSTANCE_NAME)); CREATE TABLE QRTZ_LOCKS (LOCK_NAME VARCHAR(40) NOT NULL, PRIMARY KEY (LOCK_NAME)); INSERT INTO QRTZ_LOCKS values('TRIGGER_ACCESS'); INSERT INTO QRTZ_LOCKS values('JOB_ACCESS'); INSERT INTO QRTZ_LOCKS values('CALENDAR_ACCESS'); INSERT INTO QRTZ_LOCKS values('STATE_ACCESS'); INSERT INTO QRTZ_LOCKS values('MISFIRE_ACCESS');	
PostgreSQL	CREATE TABLE QRTZ_JOB_DETAILS (JOB_NAME VARCHAR(80) NOT NULL, JOB_GROUP VARCHAR(80) NOT NULL, DESCRIPTION VARCHAR(80) NULL, JOB_CLASS_NAME VARCHAR(150) NOT NULL, IS_DURABLE BOOLEAN NOT NULL, IS_VOLATILE BOOLEAN NOT NULL, IS_STATEFUL BOOLEAN NOT NULL, REQUESTS_RECOVERY BOOLEAN NOT NULL, JOB_DATA BYTEA NULL, PRIMARY KEY (JOB_NAME,JOB_GROUP)); CREATE TABLE QRTZ_JOB_LISTENERS (JOB_NAME VARCHAR(80) NOT NULL, JOB_GROUP VARCHAR(80) NOT NULL, JOB_LISTENER VARCHAR(100) NOT NULL, PRIMARY KEY (JOB_NAME,JOB_GROUP,JOB_LISTENER), FOREIGN KEY (JOB_NAME,JOB_GROUP) REFERENCES QRTZ_JOB_DETAILS (JOB_NAME,JOB_GROUP)); CREATE TABLE QRTZ_TRIGGERS (TRIGGER_NAME VARCHAR(80) NOT NULL, TRIGGER_GROUP VARCHAR(80) NOT NULL, JOB_NAME VARCHAR(80) NOT NULL, JOB_GROUP VARCHAR(80) NOT NULL, IS_VOLATILE BOOLEAN NOT NULL, DESCRIPTION VARCHAR(100) NULL, NEXT_FIRE_TIME BIGINT NULL, PREV_FIRE_TIME BIGINT NULL, PRIORITY INTEGER NULL, TRIGGER_STATE VARCHAR(16) NOT NULL, TRIGGER_TYPE VARCHAR(8) NOT NULL, START_TIME BIGINT NOT NULL, END_TIME BIGINT NULL, CALENDAR_NAME VARCHAR(200) NULL, MISFIRE_INSTR SMALLINT NULL, JOB_DATA BYTEA NULL, PRIMARY KEY (TRIGGER_NAME,TRIGGER_GROUP), FOREIGN KEY (JOB_NAME,JOB_GROUP) REFERENCES QRTZ_JOB_DETAILS (JOB_NAME,JOB_GROUP)); CREATE TABLE QRTZ_SIMPLE_TRIGGERS (TRIGGER_NAME VARCHAR(80) NOT NULL, TRIGGER_GROUP VARCHAR(80) NOT NULL, REPEAT_COUNT BIGINT NOT NULL, REPEAT_INTERVAL BIGINT NOT NULL, TIMES_TRIGGERED	drop table QRTZ_SIMPLE_TRIGGERS; drop table QRTZ_BLOB_TRIGGERS; drop table QRTZ_CRON_TRIGGERS; drop table QRTZ_TRIGGER_LISTENERS; drop table QRTZ_CALENDARS; drop table QRTZ_FIRED_TRIGGERS; drop table QRTZ_LOCKS; drop table QRTZ_PAUSED_TRIGGER_GRPS; drop table QRTZ_SCHEDULER_STATE; drop table QRTZ_JOB_LISTENERS; drop table QRTZ_TRIGGERS; drop table QRTZ_JOB_DETAILS;

Database	Create Tables	Drop Tables
	BIGINT NOT NULL, PRIMARY KEY (TRIGGER_NAME,TRIGGER_GROUP), FOREIGN KEY (TRIGGER_NAME,TRIGGER_GROUP) REFERENCES QRTZ_TRIGGERS (TRIGGER_NAME,TRIGGER_GROUP)); CREATE TABLE QRTZ_CRON_TRIGGERS (TRIGGER_NAME VARCHAR(80) NOT NULL, TRIGGER_GROUP VARCHAR(80) NOT NULL, CRON_EXPRESSION VARCHAR(200) NOT NULL, TIME_ZONE_ID VARCHAR(80), PRIMARY KEY (TRIGGER_NAME,TRIGGER_GROUP), FOREIGN KEY (TRIGGER_NAME,TRIGGER_GROUP) REFERENCES QRTZ_TRIGGERS (TRIGGER_NAME,TRIGGER_GROUP)); CREATE TABLE QRTZ_BLOB_TRIGGERS (TRIGGER_NAME VARCHAR(80) NOT NULL, TRIGGER_GROUP VARCHAR(80) NOT NULL, BLOB_DATA BYTEA NULL, PRIMARY KEY (TRIGGER_NAME,TRIGGER_GROUP), FOREIGN KEY (TRIGGER_NAME,TRIGGER_GROUP) REFERENCES QRTZ_TRIGGERS (TRIGGER_NAME,TRIGGER_GROUP)); CREATE TABLE QRTZ_TRIGGER_LISTENERS (TRIGGER_NAME VARCHAR(80) NOT NULL, TRIGGER_GROUP VARCHAR(80) NOT NULL, TRIGGER_LISTENER VARCHAR(80) NOT NULL, PRIMARY KEY (TRIGGER_NAME,TRIGGER_GROUP,TRIGGER_LISTENER), FOREIGN KEY (TRIGGER_NAME,TRIGGER_GROUP) REFERENCES QRTZ_TRIGGERS (TRIGGER_NAME,TRIGGER_GROUP)); CREATE TABLE QRTZ_CALENDARS (CALENDAR_NAME VARCHAR(200) NOT NULL, CALENDAR BYTEA NOT NULL, PRIMARY KEY (CALENDAR_NAME)); CREATE TABLE QRTZ_PAUSED_TRIGGER_GRPS (TRIGGER_GROUP VARCHAR(200) NOT NULL, PRIMARY KEY (TRIGGER_GROUP)); CREATE TABLE QRTZ_FIRED_TRIGGERS (ENTRY_ID VARCHAR(95) NOT NULL, TRIGGER_NAME VARCHAR(80) NOT NULL, TRIGGER_GROUP VARCHAR(80) NOT NULL, IS_VOLATILE BOOLEAN NOT NULL, INSTANCE_NAME VARCHAR(200) NOT NULL, FIRED_TIME BIGINT NOT NULL, PRIORITY INTEGER NOT NULL, STATE VARCHAR(16) NOT NULL, JOB_NAME VARCHAR(200) NULL, JOB_GROUP VARCHAR(200) NULL, IS_STATEFUL BOOLEAN NULL, REQUESTS_RECOVERY BOOLEAN NULL, PRIMARY KEY (ENTRY_ID)); CREATE TABLE QRTZ_SCHEDULER_STATE	

Database	Create Tables	Drop Tables
	(INSTANCE_NAME VARCHAR(200) NOT NULL, LAST_CHECKIN_TIME BIGINT NOT NULL, CHECKIN_INTERVAL BIGINT NOT NULL, PRIMARY KEY (INSTANCE_NAME)); CREATE TABLE QRTZ_LOCKS (LOCK_NAME VARCHAR(40) NOT NULL, PRIMARY KEY (LOCK_NAME)); INSERT INTO QRTZ_LOCKS values('TRIGGER_ACCESS'); INSERT INTO QRTZ_LOCKS values('JOB_ACCESS'); INSERT INTO QRTZ_LOCKS values('CALENDAR_ACCESS'); INSERT INTO QRTZ_LOCKS values('STATE_ACCESS'); INSERT INTO QRTZ_LOCKS values('MISFIRE_ACCESS');	
Sybase	create table QRTZ_CALENDARS (CALENDAR_NAME varchar(80) not null, CALENDAR image not null) go create table QRTZ_CRON_TRIGGERS (TRIGGER_NAME varchar(80) not null, TRIGGER_GROUP varchar(80) not null, CRON_EXPRESSION varchar(120) not null, TIME_ZONE_ID varchar(80) null,) go create table QRTZ_PAUSED_TRIGGER_GRPS (TRIGGER_GROUP varchar(80) not null,) go create table QRTZ_FIRED_TRIGGERS(ENTRY_ID varchar(95) not null, TRIGGER_NAME varchar(80) not null, TRIGGER_GROUP varchar(80) not null, IS_VOLATILE bit not null, INSTANCE_NAME varchar(80) not null, FIRED_TIME numeric(13,0) not null, PRIORITY int not null, STATE varchar(16) not null, JOB_NAME varchar(80) null, JOB_GROUP varchar(80) null, IS_STATEFUL bit not null, REQUESTS_RECOVERY bit not null,) go create table QRTZ_SCHEDULER_STATE (INSTANCE_NAME varchar(80) not null, LAST_CHECKIN_TIME numeric(13,0) not null, CHECKIN_INTERVAL numeric(13,0) not null,) go create table QRTZ_LOCKS (LOCK_NAME varchar(40) not null,) go insert into QRTZ_LOCKS values('TRIGGER_ACCESS') go insert into QRTZ_LOCKS values('JOB_ACCESS') go insert into QRTZ_LOCKS values('CALENDAR_ACCESS') go insert into QRTZ_LOCKS values('STATE_ACCESS') go create table QRTZ_JOB_DETAILS (JOB_NAME varchar(80) not null, JOB_GROUP varchar(80) not null, DESCRIPTION varchar(120) null, JOB_CLASS_NAME	drop table QRTZ_SIMPLE_TRIGGERS; drop table QRTZ_BLOB_TRIGGERS; drop table QRTZ_CRON_TRIGGERS; drop table QRTZ_TRIGGER_LISTENERS; drop table QRTZ_CALENDARS; drop table QRTZ_FIRED_TRIGGERS; drop table QRTZ_LOCKS; drop table QRTZ_PAUSED_TRIGGER_GRPS; drop table QRTZ_SCHEDULER_STATE; drop table QRTZ_JOB_LISTENERS; drop table QRTZ_TRIGGERS; drop table QRTZ_JOB_DETAILS;

Database	Create Tables	Drop Tables
	varchar(128) not null, IS_DURABLE bit not null, IS_VOLATILE bit not null, IS_STATEFUL bit not null, REQUESTS_RECOVERY bit not null, JOB_DATA image null) go create table QRTZ_JOB_LISTENERS (JOB_NAME varchar(80) not null, JOB_GROUP varchar(80) not null, JOB_LISTENER varchar(80) not null) go create table QRTZ_SIMPLE_TRIGGERS (TRIGGER_NAME varchar(80) not null, TRIGGER_GROUP varchar(80) not null, REPEAT_COUNT numeric(13,0) not null, REPEAT_INTERVAL numeric(13,0) not null, TIMES_TRIGGERED numeric(13,0) not null) go create table QRTZ_BLOB_TRIGGERS (TRIGGER_NAME varchar(80) not null, TRIGGER_GROUP varchar(80) not null, BLOB_DATA image null) go create table QRTZ_TRIGGER_LISTENERS (TRIGGER_NAME varchar(80) not null, TRIGGER_GROUP varchar(80) not null, TRIGGER_LISTENER varchar(80) not null) go create table QRTZ_TRIGGERS (TRIGGER_NAME varchar(80) not null, TRIGGER_GROUP varchar(80) not null, JOB_NAME varchar(80) not null, JOB_GROUP varchar(80) not null, IS_VOLATILE bit not null, DESCRIPTION varchar(120) null, NEXT_FIRE_TIME numeric(13,0) null, PREV_FIRE_TIME numeric(13,0) null, PRIORITY int null, TRIGGER_STATE varchar(16) not null, TRIGGER_TYPE varchar(8) not null, START_TIME numeric(13,0) not null, END_TIME numeric(13,0) null, CALENDAR_NAME varchar(80) null, MISFIRE_INSTR smallint null, JOB_DATA image null) go alter table QRTZ_CALENDARS add constraint PK_qrtz_calendars primary key clustered (CALENDAR_NAME) go alter table QRTZ_CRON_TRIGGERS add constraint PK_qrtz_cron_triggers primary key clustered (TRIGGER_NAME, TRIGGER_GROUP) go alter table QRTZ_FIRED_TRIGGERS add constraint PK_qrtz_fired_triggers primary key clustered (ENTRY_ID) go alter table QRTZ_PAUSED_TRIGGER_GRPS add constraint PK_qrtz_paused_trigger_grps primary key clustered (TRIGGER_GROUP) go alter table QRTZ_SCHEDULER_STATE add constraint PK_qrtz_scheduler_state primary key clustered (INSTANCE_NAME)	

Database	Create Tables	Drop Tables
	go alter table QRTZ_LOCKS add constraint PK_qrtz_locks primary key clustered (LOCK_NAME) go alter table QRTZ_JOB_DETAILS add constraint PK_qrtz_job_details primary key clustered (JOB_NAME, JOB_GROUP) go alter table QRTZ_JOB_LISTENERS add constraint PK_qrtz_job_listeners primary key clustered (JOB_NAME, JOB_GROUP, JOB_LISTENER) go alter table QRTZ_SIMPLE_TRIGGERS add constraint PK_qrtz_simple_triggers primary key clustered (TRIGGER_NAME, TRIGGER_GROUP) go alter table QRTZ_TRIGGER_LISTENERS add constraint PK_qrtz_trigger_listeners primary key clustered (TRIGGER_NAME, TRIGGER_GROUP, TRIGGER_LISTENER) go alter table QRTZ_TRIGGERS add constraint PK_qrtz_triggers primary key clustered (TRIGGER_NAME, TRIGGER_GROUP) go alter table QRTZ_BLOB_TRIGGERS add constraint PK_qrtz_blob_triggers primary key clustered (TRIGGER_NAME, TRIGGER_GROUP) go alter table QRTZ_CRON_TRIGGERS add constraint FK_cron_triggers_triggers foreign key (TRIGGER_NAME,TRIGGER_GROUP) references QRTZ_TRIGGERS (TRIGGER_NAME,TRIGGER_GROUP) go alter table QRTZ_JOB_LISTENERS add constraint FK_job_listeners_job_details foreign key (JOB_NAME,JOB_GROUP) references QRTZ_JOB_DETAILS (JOB_NAME,JOB_GROUP) go alter table QRTZ_SIMPLE_TRIGGERS add constraint FK_simple_triggers_triggers foreign key (TRIGGER_NAME,TRIGGER_GROUP) references QRTZ_TRIGGERS (TRIGGER_NAME,TRIGGER_GROUP) go alter table QRTZ_TRIGGER_LISTENERS add constraint FK_trigger_listeners_triggers foreign key (TRIGGER_NAME,TRIGGER_GROUP) references QRTZ_TRIGGERS (TRIGGER_NAME,TRIGGER_GROUP) go alter table QRTZ_TRIGGERS add constraint FK_triggers_job_details foreign key (JOB_NAME,JOB_GROUP) references QRTZ_JOB_DETAILS (JOB_NAME,JOB_GROUP) go alter table QRTZ_BLOB_TRIGGERS add constraint FK_blob_triggers_triggers foreign key (TRIGGER_NAME,TRIGGER_GROUP)	

Database	Create Tables	Drop Tables
	references QRTZ_TRIGGERS (TRIGGER_NAME,TRIGGER_GROUP) go	

The following is a Quartz verification script.

```
select count(*) from QRTZ_SIMPLE_TRIGGERS;
select count(*) from QRTZ_BLOB_TRIGGERS;
select count(*) from QRTZ_CRON_TRIGGERS;
select count(*) from QRTZ_TRIGGER_LISTENERS;
select count(*) from QRTZ_CALENDARS;
select count(*) from QRTZ_FIRED_TRIGGERS;
select count(*) from QRTZ_LOCKS;
select count(*) from QRTZ_PAUSED_TRIGGER_GRPS;
select count(*) from QRTZ_SCHEDULER_STATE;
select count(*) from QRTZ_JOB_LISTENERS;
select count(*) from QRTZ_TRIGGERS;
select count(*) from QRTZ_JOB_DETAILS;
```

SQL Scripts for Log Database Tables

Database	Create Tables	Drop Tables
IBM DB2	CREATE TABLE ROBOT_MESSAGE (ID BIGINT NOT NULL, EXECUTIONID VARCHAR(255) NOT NULL, LOCATION CLOB(268435456), LOCATIONCODE VARCHAR(3000), MESSAGE VARCHAR(3000), MESSAGEDETAILS CLOB(268435456), PROJECTNAME VARCHAR(255), ROBOTNAME VARCHAR(255), ROBOTRUNID BIGINT NOT NULL, SEVERITY INTEGER NOT NULL, STEPNAME VARCHAR(255), TIME TIMESTAMP NOT NULL, PRIMARY KEY (ID)); CREATE INDEX INDEX_ROBOT_MESSAGE_TIME ON ROBOT_MESSAGE (TIME); CREATE TABLE ROBOT_RUN (ID BIGINT NOT NULL, BYTES BIGINT, CLIENTADDRESS VARCHAR(255), CLIENTHOST VARCHAR(255), EMAILS BIGINT, ENDTIME TIMESTAMP, ERRORCOUNT BIGINT, EXECUTIONID VARCHAR(255) NOT NULL UNIQUE, EXECUTIONTIME BIGINT, EXTRACTEDOBJECTS BIGINT, HOST VARCHAR(255), HOSTADDRESS VARCHAR(255), HTTPREQUESTS BIGINT, INFOCOUNT BIGINT, INPUTOBJECTSTRING CLOB(268435456), INPUTOBJECTS BLOB(268435456), JSINSTRUCTIONS BIGINT, KCUPOINTS BIGINT, KCUWAIT BIGINT, PROJECTNAME VARCHAR(255), QUEUETIME BIGINT, ROBOTNAME VARCHAR(255), STARTREQUESTEDTIME TIMESTAMP NOT NULL, STARTTIME TIMESTAMP, STEPS BIGINT, TOTALTIME BIGINT, WARNINGCOUNT BIGINT, PRIMARY KEY (ID)); CREATE INDEX IDX_ROBOT_RUNSTARTTIME ON ROBOT_RUN (STARTTIME); CREATE INDEX IDX_EXECUTIONID ON ROBOT_RUN (EXECUTIONID); CREATE TABLE SCHEDULE_MESSAGE (ID BIGINT NOT NULL, MESSAGE VARCHAR(3000), MESSAGEDETAILS CLOB(268435456), PROJECTNAME VARCHAR(255) NOT NULL, SCHEDULEID BIGINT NOT NULL,	DROP TABLE ROBOT_MESSAGE; DROP TABLE ROBOT_RUN; DROP TABLE SCHEDULE_MESSAGE; DROP TABLE SCHEDULE_RUN; DROP TABLE SERVER_MESSAGE; DROP SEQUENCE SEQ_GEN_SEQUENCE;

Database	Create Tables	Drop Tables
	SCHEDULENAME VARCHAR(255) NOT NULL, SCHEDULERUNID BIGINT NOT NULL, SEVERITY INTEGER NOT NULL, TIME TIMESTAMP NOT NULL, PRIMARY KEY (ID)); CREATE INDEX IDX_SCHEDULE_MSG_TIME ON SCHEDULE_MESSAGE (TIME); CREATE TABLE SCHEDULE_RUN (ID BIGINT NOT NULL, ENDTIME TIMESTAMP, ERRORCOUNT BIGINT, HOSTNAME VARCHAR(255) NOT NULL, INFOCOUNT BIGINT, PROJECTNAME VARCHAR(255) NOT NULL, QUEUETIME BIGINT, RESULT VARCHAR(255), SCHEDULEID BIGINT, SCHEDULENAME VARCHAR(255) NOT NULL, TOTALTIME BIGINT, TRIGGERTIME TIMESTAMP NOT NULL, WARNINGCOUNT BIGINT, PRIMARY KEY (ID)); CREATE INDEX IDX_SCHEDULE_TRIGTIME ON SCHEDULE_RUN (TRIGGERTIME); CREATE INDEX IDX_SCHEDULEID ON SCHEDULE_RUN (SCHEDULEID); CREATE TABLE SERVER_MESSAGE (ID BIGINT NOT NULL, HOST VARCHAR(255), HOSTADDRESS VARCHAR(255), MESSAGE VARCHAR(3500), MESSAGEDETAILS VARCHAR(3500), SEVERITY INTEGER NOT NULL, TIME TIMESTAMP NOT NULL, PRIMARY KEY (ID)); CREATE INDEX IDX_SERVER_MSG_TIME ON SERVER_MESSAGE (TIME); CREATE SEQUENCE SEQ_GEN_SEQUENCE INCREMENT BY 50 START WITH 50;	
Derby	CREATE TABLE ROBOT_MESSAGE (ID BIGINT NOT NULL, EXECUTIONID VARCHAR(255) NOT NULL, LOCATION CLOB(2147483647), LOCATIONCODE VARCHAR(5000), MESSAGE VARCHAR(5000), MESSAGEDETAILS CLOB(2147483647), PROJECTNAME VARCHAR(255), ROBOTNAME VARCHAR(255), ROBOTRUNID BIGINT NOT NULL, SEVERITY INTEGER NOT NULL, STEPNAME VARCHAR(255), TIME TIMESTAMP NOT NULL, PRIMARY KEY (ID)); CREATE INDEX INDEX_ROBOT_MESSAGE_TIME ON ROBOT_MESSAGE (TIME); CREATE TABLE ROBOT_RUN (ID BIGINT NOT NULL, BYTES BIGINT, CLIENTADDRESS VARCHAR(255), CLIENTHOST VARCHAR(255), EMAILS BIGINT, ENDTIME TIMESTAMP, ERRORCOUNT BIGINT, EXECUTIONID VARCHAR(255) NOT NULL UNIQUE, EXECUTIONTIME BIGINT, EXTRACTEDOBJECTS BIGINT, HOST VARCHAR(255), HOSTADDRESS VARCHAR(255), HTTPREQUESTS BIGINT, INFOCOUNT BIGINT, INPUTOBJECTSTRING CLOB(2147483647), INPUTOBJECTS BLOB(2147483647), JSINSTRUCTIONS BIGINT, KCUPOINTS BIGINT, KCUWAIT BIGINT, PROJECTNAME VARCHAR(255), QUEUETIME BIGINT, ROBOTNAME VARCHAR(255), STARTREQUESTEDTIME TIMESTAMP NOT NULL, STARTTIME TIMESTAMP, STEPS BIGINT, TOTALTIME BIGINT, WARNINGCOUNT BIGINT, PRIMARY KEY (ID)); CREATE INDEX IDX_ROBOT_RUNSTARTTIME ON ROBOT_RUN (STARTTIME); CREATE INDEX IDX_EXECUTIONID ON ROBOT_RUN (EXECUTIONID); CREATE TABLE SCHEDULE_MESSAGE (ID BIGINT NOT NULL,	DROP TABLE ROBOT_MESSAGE; DROP TABLE ROBOT_RUN; DROP TABLE SCHEDULE_MESSAGE; DROP TABLE SCHEDULE_RUN; DROP TABLE SERVER_MESSAGE; DROP SEQUENCE SEQ_GEN_SEQUENCE RESTRICT;

Database	Create Tables	Drop Tables
	MESSAGE VARCHAR(5000), MESSAGEDETAILS CLOB(2147483647), PROJECTNAME VARCHAR(255) NOT NULL, SCHEDULEID BIGINT NOT NULL, SCHEDULENAME VARCHAR(255) NOT NULL, SCHEDULERUNID BIGINT NOT NULL, SEVERITY INTEGER NOT NULL, TIME TIMESTAMP NOT NULL, PRIMARY KEY (ID)); CREATE INDEX IDX_SCHEDULE_MSG_TIME ON SCHEDULE_MESSAGE (TIME); CREATE TABLE SCHEDULE_RUN (ID BIGINT NOT NULL, ENDTIME TIMESTAMP, ERRORCOUNT BIGINT, HOSTNAME VARCHAR(255) NOT NULL, INFOCOUNT BIGINT, PROJECTNAME VARCHAR(255) NOT NULL, QUEUETIME BIGINT, RESULT VARCHAR(255), SCHEDULEID BIGINT, SCHEDULENAME VARCHAR(255) NOT NULL, TOTALTIME BIGINT, TRIGGERTIME TIMESTAMP NOT NULL, WARNINGCOUNT BIGINT, PRIMARY KEY (ID)); CREATE INDEX IDX_SCHEDULE_TRIGTIME ON SCHEDULE_RUN (TRIGGERTIME); CREATE INDEX IDX_SCHEDULEID ON SCHEDULE_RUN (SCHEDULEID); CREATE TABLE SERVER_MESSAGE (ID BIGINT NOT NULL, HOST VARCHAR(255), HOSTADDRESS VARCHAR(255), MESSAGE VARCHAR(5000), MESSAGEDETAILS VARCHAR(5000), SEVERITY INTEGER NOT NULL, TIME TIMESTAMP NOT NULL, PRIMARY KEY (ID)); CREATE INDEX IDX_SERVER_MSG_TIME ON SERVER_MESSAGE (TIME); CREATE SEQUENCE SEQ_GEN_SEQUENCE INCREMENT BY 50 START WITH 50;	
MySQL	CREATE TABLE ROBOT_MESSAGE (ID BIGINT AUTO_INCREMENT NOT NULL, EXECUTIONID VARCHAR(255) NOT NULL, LOCATION LONGTEXT, LOCATIONCODE VARCHAR(5000), MESSAGE VARCHAR(5000), MESSAGEDETAILS LONGTEXT, PROJECTNAME VARCHAR(255), ROBOTNAME VARCHAR(255), ROBOTRUNID BIGINT NOT NULL, SEVERITY INTEGER NOT NULL, STEPNAME VARCHAR(255), TIME DATETIME NOT NULL, PRIMARY KEY (ID)); CREATE INDEX INDEX_ROBOT_MESSAGE_TIME ON ROBOT_MESSAGE (TIME); CREATE TABLE ROBOT_RUN (ID BIGINT AUTO_INCREMENT NOT NULL, BYTES BIGINT, CLIENTADDRESS VARCHAR(255), CLIENTHOST VARCHAR(255), EMAILS BIGINT, ENDTIME DATETIME, ERRORCOUNT BIGINT, EXECUTIONID VARCHAR(255) NOT NULL UNIQUE, EXECUTIONTIME BIGINT, EXTRACTEDOBJECTS BIGINT, HOST VARCHAR(255), HOSTADDRESS VARCHAR(255), HTTPREQUESTS BIGINT, INFOCOUNT BIGINT, INPUTOBJECTSTRING LONGTEXT, INPUTOBJECTS LONGBLOB, JSINSTRUCTIONS BIGINT, KCUPOINTS BIGINT, KCUWAIT BIGINT, PROJECTNAME VARCHAR(255), QUEUETIME BIGINT, ROBOTNAME VARCHAR(255), STARTREQUESTEDTIME DATETIME NOT NULL, STARTTIME DATETIME, STEPS BIGINT, TOTALTIME BIGINT, WARNINGCOUNT BIGINT, PRIMARY KEY (ID)); CREATE INDEX	DROP TABLE ROBOT_MESSAGE; DROP TABLE ROBOT_RUN; DROP TABLE SCHEDULE_MESSAGE; DROP TABLE SCHEDULE_RUN; DROP TABLE SERVER_MESSAGE;

Database	Create Tables	Drop Tables
	IDX_ROBOT_RUNSTARTTIME ON ROBOT_RUN (STARTTIME); CREATE INDEX IDX_EXECUTIONID ON ROBOT_RUN (EXECUTIONID); CREATE TABLE SCHEDULE_MESSAGE (ID BIGINT AUTO_INCREMENT NOT NULL, MESSAGE VARCHAR(5000), MESSAGEDETAILS LONGTEXT, PROJECTNAME VARCHAR(255) NOT NULL, SCHEDULEID BIGINT NOT NULL, SCHEDULENAME VARCHAR(255) NOT NULL, SCHEDULERUNID BIGINT NOT NULL, SEVERITY INTEGER NOT NULL, TIME DATETIME NOT NULL, PRIMARY KEY (ID)); CREATE INDEX IDX_SCHEDULE_MSG_TIME ON SCHEDULE_MESSAGE (TIME); CREATE TABLE SCHEDULE_RUN (ID BIGINT AUTO_INCREMENT NOT NULL, ENDTIME DATETIME, ERRORCOUNT BIGINT, HOSTNAME VARCHAR(255) NOT NULL, INFOCOUNT BIGINT, PROJECTNAME VARCHAR(255) NOT NULL, QUEUETIME BIGINT, RESULT VARCHAR(255), SCHEDULEID BIGINT, SCHEDULENAME VARCHAR(255) NOT NULL, TOTALTIME BIGINT, TRIGGERTIME DATETIME NOT NULL, WARNINGCOUNT BIGINT, PRIMARY KEY (ID)); CREATE INDEX IDX_SCHEDULE_TRIGTIME ON SCHEDULE_RUN (TRIGGERTIME); CREATE INDEX IDX_SCHEDULEID ON SCHEDULE_RUN (SCHEDULEID); CREATE TABLE SERVER_MESSAGE (ID BIGINT AUTO_INCREMENT NOT NULL, HOST VARCHAR(255), HOSTADDRESS VARCHAR(255), MESSAGE VARCHAR(5000), MESSAGEDETAILS VARCHAR(5000), SEVERITY INTEGER NOT NULL, TIME DATETIME NOT NULL, PRIMARY KEY (ID)); CREATE INDEX IDX_SERVER_MSG_TIME ON SERVER_MESSAGE (TIME);	
Oracle	CREATE TABLE ROBOT_MESSAGE (ID NUMBER(19) NOT NULL, EXECUTIONID NVARCHAR2(255) NOT NULL, LOCATION NCLOB NULL, LOCATIONCODE NVARCHAR2(2000) NULL, MESSAGE NVARCHAR2(2000) NULL, MESSAGEDETAILS NCLOB NULL, PROJECTNAME NVARCHAR2(255) NULL, ROBOTNAME NVARCHAR2(255) NULL, ROBOTRUNID NUMBER(19) NOT NULL, SEVERITY NUMBER(10) NOT NULL, STEPNAME NVARCHAR2(255) NULL, TIME TIMESTAMP NOT NULL, PRIMARY KEY (ID)); CREATE INDEX INDEX_ROBOT_MESSAGE_TIME ON ROBOT_MESSAGE (TIME); CREATE TABLE ROBOT_RUN (ID NUMBER(19) NOT NULL, BYTES NUMBER(19) NULL, CLIENTADDRESS NVARCHAR2(255) NULL, CLIENTHOST NVARCHAR2(255) NULL, EMAILS NUMBER(19) NULL, ENDTIME TIMESTAMP NULL, ERRORCOUNT NUMBER(19) NULL, EXECUTIONID NVARCHAR2(255) NOT NULL UNIQUE, EXECUTIONTIME NUMBER(19) NULL, EXTRACTEDOBJECTS NUMBER(19) NULL, HOST NVARCHAR2(255) NULL, HOSTADDRESS NVARCHAR2(255) NULL, HTTPREQUESTS NUMBER(19) NULL, INFOCOUNT NUMBER(19) NULL, INPUTOBJECTSTRING NCLOB NULL, INPUTOBJECTS BLOB NULL, JSINSTRUCTIONS NUMBER(19) NULL, KCUPOINTS NUMBER(19)	DROP TABLE ROBOT_MESSAGE CASCADE CONSTRAINTS; DROP TABLE ROBOT_RUN CASCADE CONSTRAINTS; DROP TABLE SCHEDULE_MESSAGE CASCADE CONSTRAINTS; DROP TABLE SCHEDULE_RUN CASCADE CONSTRAINTS; DROP TABLE SERVER_MESSAGE CASCADE CONSTRAINTS; DROP SEQUENCE SEQ_GEN_SEQUENCE;

Database	Create Tables	Drop Tables
	NULL, KCUWAIT NUMBER(19) NULL, PROJECTNAME NVARCHAR2(255) NULL, QUEUETIME NUMBER(19) NULL, ROBOTNAME NVARCHAR2(255) NULL, STARTREQUESTEDTIME TIMESTAMP NOT NULL, STARTTIME TIMESTAMP NULL, STEPS NUMBER(19) NULL, TOTALTIME NUMBER(19) NULL, WARNINGCOUNT NUMBER(19) NULL, PRIMARY KEY (ID)); CREATE INDEX IDX_ROBOT_RUNSTARTTIME ON ROBOT_RUN (STARTTIME); CREATE INDEX IDX_EXECUTIONID ON ROBOT_RUN (EXECUTIONID); CREATE TABLE SCHEDULE_MESSAGE (ID NUMBER(19) NOT NULL, MESSAGE NVARCHAR2(2000) NULL, MESSAGEDETAILS NCLOB NULL, PROJECTNAME NVARCHAR2(255) NOT NULL, SCHEDULEID NUMBER(19) NOT NULL, SCHEDULENAME NVARCHAR2(255) NOT NULL, SCHEDULERUNID NUMBER(19) NOT NULL, SEVERITY NUMBER(10) NOT NULL, TIME TIMESTAMP NOT NULL, PRIMARY KEY (ID)); CREATE INDEX IDX_SCHEDULE_MSG_TIME ON SCHEDULE_MESSAGE (TIME); CREATE TABLE SCHEDULE_RUN (ID NUMBER(19) NOT NULL, ENDTIME TIMESTAMP NULL, ERRORCOUNT NUMBER(19) NULL, HOSTNAME NVARCHAR2(255) NOT NULL, INFOCOUNT NUMBER(19) NULL, PROJECTNAME NVARCHAR2(255) NOT NULL, QUEUETIME NUMBER(19) NULL, RESULT NVARCHAR2(255) NULL, SCHEDULEID NUMBER(19) NULL, SCHEDULENAME NVARCHAR2(255) NOT NULL, TOTALTIME NUMBER(19) NULL, TRIGGERTIME TIMESTAMP NOT NULL, WARNINGCOUNT NUMBER(19) NULL, PRIMARY KEY (ID)); CREATE INDEX IDX_SCHEDULE_TRIGTIME ON SCHEDULE_RUN (TRIGGERTIME); CREATE INDEX IDX_SCHEDULEID ON SCHEDULE_RUN (SCHEDULEID); CREATE TABLE SERVER_MESSAGE (ID NUMBER(19) NOT NULL, HOST NVARCHAR2(255) NULL, HOSTADDRESS NVARCHAR2(255) NULL, MESSAGE NVARCHAR2(2000) NULL, MESSAGEDETAILS NVARCHAR2(2000) NULL, SEVERITY NUMBER(10) NOT NULL, TIME TIMESTAMP NOT NULL, PRIMARY KEY (ID)); CREATE INDEX IDX_SERVER_MSG_TIME ON SERVER_MESSAGE (TIME); CREATE SEQUENCE SEQ_GEN_SEQUENCE INCREMENT BY 50 START WITH 50;	
PostgreSQL	CREATE TABLE ROBOT_MESSAGE (ID BIGINT NOT NULL, EXECUTIONID VARCHAR(255) NOT NULL, LOCATION TEXT, LOCATIONCODE VARCHAR(5000), MESSAGE VARCHAR(5000), MESSAGEDETAILS TEXT, PROJECTNAME VARCHAR(255), ROBOTNAME VARCHAR(255), ROBOTRUNID BIGINT NOT NULL, SEVERITY INTEGER NOT NULL, STEPNAME VARCHAR(255), TIME TIMESTAMP NOT NULL, PRIMARY KEY (ID)); CREATE INDEX INDEX_ROBOT_MESSAGE_TIME ON ROBOT_MESSAGE (TIME); CREATE TABLE ROBOT_RUN (ID BIGINT NOT NULL, BYTES BIGINT, CLIENTADDRESS VARCHAR(255), CLIENTHOST VARCHAR(255), EMAILS BIGINT, ENDTIME TIMESTAMP, ERRORCOUNT BIGINT, EXECUTIONID	DROP TABLE ROBOT_MESSAGE CASCADE; DROP TABLE ROBOT_RUN CASCADE; DROP TABLE SCHEDULE_MESSAGE CASCADE; DROP TABLE SCHEDULE_RUN CASCADE; DROP TABLE SERVER_MESSAGE CASCADE;

Database	Create Tables	Drop Tables
	VARCHAR(255) NOT NULL UNIQUE, EXECUTIONTIME BIGINT, EXTRACTEDOBJECTS BIGINT, HOST VARCHAR(255), HOSTADDRESS VARCHAR(255), HTTPREQUESTS BIGINT, INFOCOUNT BIGINT, INPUTOBJECTSTRING TEXT, INPUTOBJECTS BYTEA, JSINSTRUCTIONS BIGINT, KCUPOINTS BIGINT, KCUWAIT BIGINT, PROJECTNAME VARCHAR(255), QUEUETIME BIGINT, ROBOTNAME VARCHAR(255), STARTREQUESTEDTIME TIMESTAMP NOT NULL, STARTTIME TIMESTAMP, STEPS BIGINT, TOTALTIME BIGINT, WARNINGCOUNT BIGINT, PRIMARY KEY (ID)); CREATE INDEX IDX_ROBOT_RUNSTARTTIME ON ROBOT_RUN (STARTTIME); CREATE INDEX IDX_EXECUTIONID ON ROBOT_RUN (EXECUTIONID); CREATE TABLE SCHEDULE_MESSAGE (ID BIGINT NOT NULL, MESSAGE VARCHAR(5000), MESSAGEDETAILS TEXT, PROJECTNAME VARCHAR(255) NOT NULL, SCHEDULEID BIGINT NOT NULL, SCHEDULENAME VARCHAR(255) NOT NULL, SCHEDULERUNID BIGINT NOT NULL, SEVERITY INTEGER NOT NULL, TIME TIMESTAMP NOT NULL, PRIMARY KEY (ID)); CREATE INDEX IDX_SCHEDULE_MSG_TIME ON SCHEDULE_MESSAGE (TIME); CREATE TABLE SCHEDULE_RUN (ID BIGINT NOT NULL, ENDTIME TIMESTAMP, ERRORCOUNT BIGINT, HOSTNAME VARCHAR(255) NOT NULL, INFOCOUNT BIGINT, PROJECTNAME VARCHAR(255) NOT NULL, QUEUETIME BIGINT, RESULT VARCHAR(255), SCHEDULEID BIGINT, SCHEDULENAME VARCHAR(255) NOT NULL, TOTALTIME BIGINT, TRIGGERTIME TIMESTAMP NOT NULL, WARNINGCOUNT BIGINT, PRIMARY KEY (ID)); CREATE INDEX IDX_SCHEDULE_TRIGTIME ON SCHEDULE_RUN (TRIGGERTIME); CREATE INDEX IDX_SCHEDULEID ON SCHEDULE_RUN (SCHEDULEID); CREATE TABLE SERVER_MESSAGE (ID BIGINT NOT NULL, HOST VARCHAR(255), HOSTADDRESS VARCHAR(255), MESSAGE VARCHAR(5000), MESSAGEDETAILS VARCHAR(5000), SEVERITY INTEGER NOT NULL, TIME TIMESTAMP NOT NULL, PRIMARY KEY (ID)); CREATE INDEX IDX_SERVER_MSG_TIME ON SERVER_MESSAGE (TIME); CREATE SEQUENCE SEQ_GEN_SEQUENCE INCREMENT BY 50 START WITH 50;	DROP SEQUENCE SEQ_GEN_SEQUENCE;
Microsoft SQL Server	CREATE TABLE ROBOT_MESSAGE (ID NUMERIC(19) IDENTITY NOT NULL, EXECUTIONID NVARCHAR(255) NOT NULL, LOCATION NTEXT NULL, LOCATIONCODE NVARCHAR(4000) NULL, MESSAGE NVARCHAR(4000) NULL, MESSAGEDETAILS NTEXT NULL, PROJECTNAME NVARCHAR(255) NULL, ROBOTNAME NVARCHAR(255) NULL, ROBOTRUNID NUMERIC(19) NOT NULL, SEVERITY INTEGER NOT NULL, STEPNAME NVARCHAR(255) NULL, TIME DATETIME NOT NULL, PRIMARY KEY (ID)); CREATE INDEX INDEX_ROBOT_MESSAGE_TIME ON ROBOT_MESSAGE (TIME); CREATE TABLE ROBOT_RUN (ID NUMERIC(19) IDENTITY NOT NULL, BYTES NUMERIC(19) NULL, CLIENTADDRESS NVARCHAR(255)	DROP TABLE ROBOT_MESSAGE; DROP TABLE ROBOT_RUN; DROP TABLE SCHEDULE_MESSAGE; DROP TABLE SCHEDULE_RUN; DROP TABLE SERVER_MESSAGE;

Database	Create Tables	Drop Tables
	NULL, CLIENTHOST NVARCHAR(255) NULL, EMAILS NUMERIC(19) NULL, ENDTIME DATETIME NULL, ERRORCOUNT NUMERIC(19) NULL, EXECUTIONID NVARCHAR(255) NOT NULL UNIQUE, EXECUTIONTIME NUMERIC(19) NULL, EXTRACTEDOBJECTS NUMERIC(19) NULL, HOST NVARCHAR(255) NULL, HOSTADDRESS NVARCHAR(255) NULL, HTTPREQUESTS NUMERIC(19) NULL, INFOCOUNT NUMERIC(19) NULL, INPUTOBJECTSTRING NTEXT NULL, INPUTOBJECTS IMAGE NULL, JSINSTRUCTIONS NUMERIC(19) NULL, KCUPOINTS NUMERIC(19) NULL, KCUWAIT NUMERIC(19) NULL, PROJECTNAME NVARCHAR(255) NULL, QUEUETIME NUMERIC(19) NULL, ROBOTNAME NVARCHAR(255) NULL, STARTREQUESTEDTIME DATETIME NOT NULL, STARTTIME DATETIME NULL, STEPS NUMERIC(19) NULL, TOTALTIME NUMERIC(19) NULL, WARNINGCOUNT NUMERIC(19) NULL, PRIMARY KEY (ID)); CREATE INDEX IDX_ROBOT_RUNSTARTTIME ON ROBOT_RUN (STARTTIME); CREATE INDEX IDX_EXECUTIONID ON ROBOT_RUN (EXECUTIONID); CREATE TABLE SCHEDULE_MESSAGE (ID NUMERIC(19) IDENTITY NOT NULL, MESSAGE NVARCHAR(4000) NULL, MESSAGEDETAILS NTEXT NULL, PROJECTNAME NVARCHAR(255) NOT NULL, SCHEDULEID NUMERIC(19) NOT NULL, SCHEDULENAME NVARCHAR(255) NOT NULL, SCHEDULERUNID NUMERIC(19) NOT NULL, SEVERITY INTEGER NOT NULL, TIME DATETIME NOT NULL, PRIMARY KEY (ID)); CREATE INDEX IDX_SCHEDULE_MSG_TIME ON SCHEDULE_MESSAGE (TIME); CREATE TABLE SCHEDULE_RUN (ID NUMERIC(19) IDENTITY NOT NULL, ENDTIME DATETIME NULL, ERRORCOUNT NUMERIC(19) NULL, HOSTNAME NVARCHAR(255) NOT NULL, INFOCOUNT NUMERIC(19) NULL, PROJECTNAME NVARCHAR(255) NOT NULL, QUEUETIME NUMERIC(19) NULL, RESULT NVARCHAR(255) NULL, SCHEDULEID NUMERIC(19) NULL, SCHEDULENAME NVARCHAR(255) NOT NULL, TOTALTIME NUMERIC(19) NULL, TRIGGERTIME DATETIME NOT NULL, WARNINGCOUNT NUMERIC(19) NULL, PRIMARY KEY (ID)); CREATE INDEX IDX_SCHEDULE_TRIGTIME ON SCHEDULE_RUN (TRIGGERTIME); CREATE INDEX IDX_SCHEDULEID ON SCHEDULE_RUN (SCHEDULEID); CREATE TABLE SERVER_MESSAGE (ID NUMERIC(19) IDENTITY NOT NULL, HOST NVARCHAR(255) NULL, HOSTADDRESS NVARCHAR(255) NULL, MESSAGE NVARCHAR(4000) NULL, MESSAGEDETAILS NVARCHAR(4000) NULL, SEVERITY INTEGER NOT NULL, TIME DATETIME NOT NULL, PRIMARY KEY (ID)); CREATE INDEX IDX_SERVER_MSG_TIME ON SERVER_MESSAGE (TIME);	
Sybase	CREATE TABLE ROBOT_MESSAGE (ID NUMERIC(19) IDENTITY NOT NULL, EXECUTIONID NVARCHAR(255) NOT NULL, LOCATION TEXT NULL, LOCATIONCODE NVARCHAR(3150) NULL, MESSAGE NVARCHAR(3150) NULL, MESSAGEDETAILS TEXT NULL, PROJECTNAME NVARCHAR(255) NULL, ROBOTNAME NVARCHAR(255)	DROP TABLE ROBOT_MESSAGE; DROP TABLE ROBOT_RUN; DROP TABLE SCHEDULE_MESSAGE;

Database	Create Tables	Drop Tables
	NULL, ROBOTRUNID NUMERIC(19) NOT NULL, SEVERITY INTEGER NOT NULL, STEPNAME NVARCHAR(255) NULL, TIME DATETIME NOT NULL, PRIMARY KEY (ID)); CREATE INDEX INDEX_ROBOT_MESSAGE_TIME ON ROBOT_MESSAGE (TIME); CREATE TABLE ROBOT_RUN (ID NUMERIC(19) IDENTITY NOT NULL, BYTES NUMERIC(19) NULL, CLIENTADDRESS NVARCHAR(255) NULL, CLIENTHOST NVARCHAR(255) NULL, EMAILS NUMERIC(19) NULL, ENDTIME DATETIME NULL, ERRORCOUNT NUMERIC(19) NULL, EXECUTIONID NVARCHAR(255) NOT NULL UNIQUE, EXECUTIONTIME NUMERIC(19) NULL, EXTRACTEDOBJECTS NUMERIC(19) NULL, HOST NVARCHAR(255) NULL, HOSTADDRESS NVARCHAR(255) NULL, HTTPREQUESTS NUMERIC(19) NULL, INFOCOUNT NUMERIC(19) NULL, INPUTOBJECTSTRING TEXT NULL, INPUTOBJECTS IMAGE NULL, JSINSTRUCTIONS NUMERIC(19) NULL, KCUPOINTS NUMERIC(19) NULL, KCUWAIT NUMERIC(19) NULL, PROJECTNAME NVARCHAR(255) NULL, QUEUETIME NUMERIC(19) NULL, ROBOTNAME NVARCHAR(255) NULL, STARTREQUESTEDTIME DATETIME NOT NULL, STARTTIME DATETIME NULL, STEPS NUMERIC(19) NULL, TOTALTIME NUMERIC(19) NULL, WARNINGCOUNT NUMERIC(19) NULL, PRIMARY KEY (ID)); CREATE INDEX IDX_ROBOT_RUNSTARTTIME ON ROBOT_RUN (STARTTIME); CREATE INDEX IDX_EXECUTIONID ON ROBOT_RUN (EXECUTIONID); CREATE TABLE SCHEDULE_MESSAGE (ID NUMERIC(19) IDENTITY NOT NULL, MESSAGE NVARCHAR(3150) NULL, MESSAGEDETAILS TEXT NULL, PROJECTNAME NVARCHAR(255) NOT NULL, SCHEDULEID NUMERIC(19) NOT NULL, SCHEDULENAME NVARCHAR(255) NOT NULL, SCHEDULERUNID NUMERIC(19) NOT NULL, SEVERITY INTEGER NOT NULL, TIME DATETIME NOT NULL, PRIMARY KEY (ID)); CREATE INDEX IDX_SCHEDULE_MSG_TIME ON SCHEDULE_MESSAGE (TIME); CREATE TABLE SCHEDULE_RUN (ID NUMERIC(19) IDENTITY NOT NULL, ENDTIME DATETIME NULL, ERRORCOUNT NUMERIC(19) NULL, HOSTNAME NVARCHAR(255) NOT NULL, INFOCOUNT NUMERIC(19) NULL, PROJECTNAME NVARCHAR(255) NOT NULL, QUEUETIME NUMERIC(19) NULL, RESULT NVARCHAR(255) NULL, SCHEDULEID NUMERIC(19) NULL, SCHEDULENAME NVARCHAR(255) NOT NULL, TOTALTIME NUMERIC(19) NULL, TRIGGERTIME DATETIME NOT NULL, WARNINGCOUNT NUMERIC(19) NULL, PRIMARY KEY (ID)); CREATE INDEX IDX_SCHEDULE_TRIGTIME ON SCHEDULE_RUN (TRIGGERTIME); CREATE INDEX IDX_SCHEDULEID ON SCHEDULE_RUN (SCHEDULEID); CREATE TABLE SERVER_MESSAGE (ID NUMERIC(19) IDENTITY NOT NULL, HOST NVARCHAR(255) NULL, HOSTADDRESS NVARCHAR(255) NULL, MESSAGE NVARCHAR(3200) NULL, MESSAGEDETAILS NVARCHAR(3200) NULL, SEVERITY INTEGER NOT NULL, TIME DATETIME	DROP TABLE SCHEDULE_RUN; DROP TABLE SERVER_MESSAGE;

Database	Create Tables	Drop Tables
	NOT NULL, PRIMARY KEY (ID)); CREATE INDEX IDX_SERVER_MSG_TIME ON SERVER_MESSAGE (TIME);	

Index