

Oracle® Endeca Web Acquisition Toolkit

Katalyst Installation Reference

Version 3.2.0 • January 2016

Copyright and disclaimer

Copyright © 2003, 2016, Oracle and/or its affiliates. All rights reserved.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners. UNIX is a registered trademark of The Open Group.

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

This software or hardware and documentation may provide access to or information on content, products and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services.

Contents

Introduction	6
Additional Resources	6
Getting Help for Kofax Products	6
System Requirements	8
Supported Platforms	9
Installing Kapow	12
Installing on Windows	13
Installing on Linux	13
Silent Installation on Windows	13
Important Folders in Kapow Katalyst	14
Providing License Information	16
Management Console (License Server)	16
Design Studio	16
Kapow Katalyst Initial Configuration	20
Starting Servers Automatically	20
Starting Servers on Windows	20
Starting Servers on Linux	22
Upgrading Kapow	24
Customers Upgrading to 9.6	25
Customers Upgrading to 9.5	26
Customers Upgrading From 9.2	26
Data Migration Tutorial	27
New Excel Capabilities in 9.2	29
RoboRunner Discontinued in 9.1	30
Scheduling concepts	30
Bulk deploying to Management Console	31
Robots Schedule	31
Changes in 8.2	31
New Variables and Type System in 8.1	32
New Control Flow and Error Handling in 8.0	33
Error Handling	35
New Storage System in 7.2	37
The Old Storage System	37
The New Storage System	38
Legacy Options	38
Backward Compatibility	38

Migration	39
Index	42

Legal Notice

© 2007-2015 Kofax, Inc., 15211 Laguna Canyon Road, Irvine, California 92618, U.S.A. All rights reserved. Use is subject to license terms.

Copyright (C) 2004- 2015 Kapow Technologies, Inc. All rights reserved. Use is subject to license terms. Unauthorized duplication or distribution is strictly prohibited.

Third-party software is copyrighted and licensed from Kofax's suppliers. For information on third-party software included in this product, see [documentation/thirdparty.html](#) located in your Kapow installation folder.

This product is protected by U.S. Patent No. 6,370,277.

THIS SOFTWARE CONTAINS CONFIDENTIAL INFORMATION AND TRADE SECRETS OF KOFAX, INC. USE, DISCLOSURE OR REPRODUCTION IS PROHIBITED WITHOUT THE PRIOR EXPRESS WRITTEN PERMISSION OF KOFAX, INC.

Kofax, the Kofax logo, and the Kofax product names stated herein are trademarks or registered trademarks of Kofax, Inc. in the U.S. and other countries. All other trademarks are the trademarks or registered trademarks of their respective owners.

U.S. Government Rights Commercial software. Government users are subject to the Kofax, Inc. standard license agreement and applicable provisions of the FAR and its supplements.

You agree that you do not intend to and will not, directly or indirectly, export or transmit the Software or related documentation and technical data to any country to which such export or transmission is restricted by any applicable U.S. regulation or statute, without the prior written consent, if required, of the Bureau of Export Administration of the U.S. Department of Commerce, or such other governmental entity as may have jurisdiction over such export or transmission. You represent and warrant that you are not located in, under the control of, or a national or resident of any such country.

DOCUMENTATION IS PROVIDED "AS IS" AND ALL EXPRESS OR IMPLIED CONDITIONS, REPRESENTATIONS AND WARRANTIES, INCLUDING ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT, ARE DISCLAIMED, EXCEPT TO THE EXTENT THAT SUCH DISCLAIMERS ARE HELD TO BE LEGALLY INVALID.

Introduction

This guide covers both [interactive](#) and "[silent](#)" installation on Windows as well as normal and headless installation on [Linux](#).

Before installing on Windows or Linux, you must decide whether to install the 32-bit or the 64-bit version. The key issues to consider are:

- Your operating system may only support one of the versions. 64-bit Windows support both and on Linux it is recommended to use the one that match your installation.
- The 64-bit version is mainly useful on servers where you want to run RoboServer that use much RAM. (You will need to configure the allowed amount of RAM after installation as described in the Changing the RAM Allocation topic in the Administrator's Guide).

It is possible to install both the 32-bit and the 64-bit versions of Kofax Kapow on the same computer, and they will share configuration and robots without any extra work on your side. However, every time you want to start one of the Kofax Kapow applications, you must take care to start the right version (32-bit or 64-bit). Additionally, on Linux, make sure to select different installation folders for the 32-bit and 64-bit versions (on Windows, this is automatic).

The following subsections describe:

- The supported platforms and system requirements.
- How to install Kapow Katalyst, either interactively or "silently" (i.e., without user interaction).
- How to enter license information so that you can start Kapow Katalyst.
- How to configure Kapow Katalyst.
- How to set up Kapow Katalyst server applications to start automatically when the computer reboots.
- How to upgrade Kapow Katalyst.

Additional Resources

This section gives information about resources that are available to assist you in using .

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.

System Requirements

The tables below specify system specification for different platforms. The requirements may depend on the application so these should only be taken as guidelines and not as absolute numbers. A complex clipping solution might require much more power than a simple collection solution. The recommendations for servers are for one server. The number of servers used for a given application (the size of a cluster) is a completely different matter and should be estimated using methods described elsewhere.

IDE requirements:

	Minimum	Recommended
Windows	Intel Core Duo 1.8 GHz CPU (or AMD equivalent), 4GB RAM, 500MB Free Disk Space	Intel Core Duo 2.66 GHz CPU (or AMD equivalent), 8GB RAM, 500MB Free Disk Space
Linux	Intel Core Duo 1.8 GHz CPU (or AMD equivalent), 4GB RAM, 500MB Free Disk Space	Intel Core Duo 2.66 GHz CPU (or AMD equivalent), 8GB RAM, 500MB Free Disk Space

Server requirements:

	Minimum	Recommended
Windows (32-bit)	Intel Xeon L5520 CPU (or equivalent AMD Opteron), 2GB RAM, 500MB Free Disk Space	Intel Xeon X5680/X5677 CPU (or equivalent AMD Opteron), 2GB RAM, 500MB Free Disk Space
Linux (32-bit)	Intel Xeon L5520 CPU (or equivalent AMD Opteron), 2GB RAM, 500MB Free Disk Space	Intel Xeon X5680/X5677 CPU (or equivalent AMD Opteron), 2GB RAM, 500MB Free Disk Space
Windows (64-bit)	Intel Xeon L5520 CPU (or equivalent AMD Opteron), 4GB RAM, 500MB Free Disk Space	Intel Xeon X5680/X5677 CPU (or equivalent AMD Opteron), 8GB RAM, 500MB Free Disk Space
Linux (64-bit)	Intel Xeon L5520 CPU (or equivalent AMD Opteron), 4GB RAM, 500MB Free Disk Space	Intel Xeon X5680/X5677 CPU (or equivalent AMD Opteron), 8GB RAM, 500MB Free Disk Space

Real-time data: If you have a solution where users are waiting for results in real-time, CPU speed is normally the bottleneck, and you should buy the fastest CPU available for your hardware platform.

Dedicated hardware: For best performance we recommend that you always run RoboServer on dedicated hardware. That means that you should not run database servers and other services on the same hardware as your RoboServer.

Operating system: Kapow recommends using the Linux operating system as it is currently faster than Windows when running Java applications. In some scenarios we have measured up to 33% more throughput when using Linux.

Supported Platforms

This is the list of supported platforms. The versions listed are the ones on which Kapow Katalyst has been tested. This does not mean that it will not work on other versions. If in doubt whether it works on a version not listed, contact Kapow Software to get more information.

IDE Platforms	
Windows	Windows Server 2008 R2, Windows Server 2012, Windows Server 2012 R2, Windows Vista, Windows 7, Windows 8
Linux	CentOS/Red Hat Enterprise Linux 6.x, Ubuntu 12.04 LTS

Server Platforms	
Windows	Windows Server 2008 R2, Windows Server 2012, Windows Server 2012 R2, Windows Vista, Windows 7, Windows 8
Linux	CentOS/Red Hat Enterprise Linux 5.6 or 6.x, Debian 7.2

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.

You must also install fonts on a headless Linux server for the webkit robots to work.

- [Instructions for installing fonts for CentOS / RedHat](#)
- [Instructions for installing fonts for Ubuntu](#)

Databases	
Oracle*	Version 11g
IBM DB2	9.7

Databases	
Microsoft SQL Server	Version 2005, 2008 and 2008 R2, 2012, 2014
Sybase*	Adaptive Server Enterprise 15.0 using the jConnect for JDBC 3.0 version 6.0.5 driver
MySQL*	Version 5.1 or later
HBase	The step Store in HBase Table can be used with HBase Version 0.90.6.

Note You can use PostgreSQL database only for storing your data. PostgreSQL cannot be used as a Management Console system or logging database.

APIs	
Java	Java 8 including J2EE servers such as Oracle WebLogic 11, IBM WebSphere, and JBOSS
JMX	Java 1.6
.NET	C#, .NET Version 4.0

For correct display of data in the Kofax Insight Dashboard, make sure Java correctly sets the time according to your time zone on RoboServers and computers running Management Consoles. See the *Timezone Data Versions in the JRE Software* on the Oracle web site for the latest updates in time zones. If necessary, use the *Timezone Updater Tool* to update the time zone information.

Management Console Portals	
Tomcat	Version 7.0.56 or later. Minimum Java version 1.8 with the latest update.

Other Supported Technologies	
Supported browsers for Management Console and KappZone/Kapplets	Internet Explorer version 9, 10, and 11; Chrome version 31 and later; Firefox version 25 and later; Safari version 7.

* Be aware that loss of data may occur when storing data in Oracle, Sybase or MySQL. On Oracle empty string is converted to null. On Sybase empty string is converted to " " (a single space). On MySQL millisecond precision is lost when storing dates. For details check ObjectKey Caveats section in the Storing Data in Databases topic in Kapow help.

Installing Kapow

Kapow Katalyst has one unified installer, which installs all the software, even though different parts of it will be run on servers and on client computers. The role (server, client, etc.) of a particular installation is decided during the steps that take place after basic installation of the software.

Upgrading from Earlier Versions

In the interests of stability, different versions of Kapow Katalyst may be installed side by side on the same computer without interfering with each other (except that they must be configured to use different port numbers for Management Console if run simultaneously). This means that you can install a newer version and get acquainted with it while still doing your daily work with the older version.

This also means that important data like uploaded robots, execution schedules and so on will not be shared between different versions installed on the same computer. Thus at some point you will need to copy this data from the old Kapow Katalyst installation to the new one. (There is a similar but separate issue with Design Studio projects as discussed in Notes for Users of Earlier Versions.)

You copy this data from one version of Management Console to another by making a backup of the old installation and "restoring" it into the new one. This process is discussed in the Options tab topic. Making a backup is done slightly differently in different versions of Kapow Katalyst, so check the online documentation for the relevant old version to learn how to create a backup.

If you are running Management Console in a Tomcat web server (see below), be sure to make the backup before modifying your Tomcat configuration for the new version of Kapow Katalyst.

Management Console in Tomcat

The Management Console is a key part of Kapow Katalyst that by default is run as an embedded component inside the RoboServer application. However, it may also be installed as a regular web application on a standalone Tomcat web server. If you want to do that (or decide later to switch to doing that), you need to do a number of additional manual steps as described in the Management Console on Tomcat section in the Administrator's Guide.

When running embedded in the RoboServer application, Management Console can use an embedded derby database. At customer sites we have seen degraded performance when having more than 100 schedulers or 500 robots in the embedded Management Console. If you experience that we recommend deploying Management Console in Tomcat where you can choose another database.

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.

If you are upgrading from a previous version of Kapow Katalyst where you have installed Management Console in a Tomcat web server, be sure to make a backup of it (see above) before you install the new version into Tomcat.

Installing on Windows

You will have received instructions on how to download the installer. Download and save the KapowKatalyst_9.6.2.msi (32-bit) or KapowKatalyst_9.6.2_x64.msi (64-bit) file to your hard disk. After the download is completed, run the file to start the installation and follow the steps and dialogs of the installer. The proceed to enter license information as described in [Providing License Information](#).

Note You need to have administrator rights to install Kapow on Windows.

Installing on Linux

You will have received instructions on how to download the appropriate file. Download and save the installation .tar.gz file to your hard disk. This file is named KapowKatalyst_9.6.2.tar.gz for the Linux 32-bit version and KapowKatalyst_9.6.2_x64.tar.gz for the Linux 64-bit version. The following instructions refer to the Linux 32-bit version, but the others are the same except for the filename.

When the download is completed, the installation is done by extracting the contents of the file. In most Linux distributions, this can be done by right-clicking the file and choosing the appropriate extraction option. The file can also be extracted from the command line as follows:

```
$tar xzf KapowKatalyst_9.6.2.tar.gz
```

Alternatively, to extract the file to a specific directory, the following command can be used:

```
$tar xzf KapowKatalyst_9.6.2.tar.gz -C /destination_directory
```

When the file has been extracted, proceed to enter license information as described in [Providing License Information](#).

Note You can install as an unprivileged user on Linux.

Silent Installation on Windows

A silent installer is able to run without user interaction. This is convenient if, for instance, you need to automate the installation process in a script. If you wish to perform a silent installation of Kapow Katalyst, you need to run the following command with administrative rights:

```
msiexec /qn /i KapowKatalyst_9.6.2.msi
```

This will install the program to the default location. If you wish to specify another location, instead run:

```
msiexec /qn /i KapowKatalyst_9.6.2.msi INSTALLDIR="dir"
```

where dir is the location you wish to install to. After installation, you still need to proceed to enter license information as described in [Providing License Information](#).

Important Folders in Kapow Katalyst

There are a few folders and files in the installation that you need to know about.

Installation Folder

The installation folder is the folder where Kapow Katalyst is installed. On Windows, the installation folder defaults to:

C:\Program Files\Kapow 9.6.2

On Linux it will be a directory named KapowKatalyst_9.6.2 in the directory where you extracted the archive.

The installation folder contains the following important folders:

bin

contains all executable programs for Kapow Katalyst.

API

contains files and documentation related to the Kapow Katalyst Integration APIs.

lib/jdbc

contains the installed JDBC database drivers. These are drivers that will always be available in Kapow Katalyst applications. Normally, you should manage JDBC drivers as described in the Database Drivers topic in the Management Console section of the online help.

Project Folder

The project folder contains your Library of robots and types, as described in the Design Studio chapter of the online help. The location of the project folder can be configured with the Settings application as described in the Working with Projects and Libraries topic of the Design Studio help. On Windows, the default location is like one of the following, depending on your Windows version:

C:\Documents and Settings\ username\My Documents\My Robots\9.6.2

C:\Users\ username\Documents\My Robots\9.6.2

On Linux the default location of the project folder is:

~/KapowKatalyst/9.6.2

The project folder must contain a single folder named Library.

Application Data Folder

The application data folder contains files that are private to the Kapow Katalyst but which differ for different users of the same computer. On Windows, the application data folder is (depending on your Windows version):

C:\Documents and Settings\ username\Local Settings\Application Data\KapowKatalyst\9.6.2

C:\Users\ username\AppData\Local\KapowKatalyst\9.6.2

On Linux the application data folder is:

~/ .KapowKatalyst/9.6.2

To change the location of the application data folder you can edit the common.conf file in the bin folder of the installation folder. You should add these two lines:

```
wrapper.java.additional.10=-Dkapow.applicationDataFolder="Folder containing
Configuration folder"
```

```
wrapper.java.additional.10.stripquotes=TRUE
```

Make sure that the user running Kapow Katalyst has read and write access to the folder.

Under normal circumstances, you should never modify or delete files or folders within this folder directly; the GUI tools should be used instead. The application data folder contains the following important folders:

Certificates

contains the HTTPS certificates known by Kapow Katalyst. See the Certificates section in the Administrator's Guide for more information.

Configuration

contains configuration files, which are maintained as described in the section [Configuring Kapow Katalyst](#).

Data

contains the embedded Derby database used by the Management Console (except when installed as a web application in a Tomcat web server). See Management Console on Tomcat chapter in the Administrator's Guide.

DemoDatabase

contains the Development Database which is used in the Management Console Beginner Tutorial, and which you can also use in "toy projects" during your introduction to Kapow Katalyst.

Logs

contains log files.

Providing License Information

The first step after basic installation is to enter license information. See the following pages for information on where to enter a license key in Management Console and Design Studio, respectively. It is not necessary to enter license information in the RoboServers, as it will automatically receive the necessary license information from Management Console.

Management Console (License Server)

Before you can enter license information into Management Console, you need to start it.

Windows

Use the **Start Management Console** item in the Start menu.

Linux

Start Management Console from the command line. It is part of the RoboServer program, which is found in the bin directory under the installation (see Important Folders in Kapow Katalyst). It can be started with:

```
$. /RoboServer -p 50000 -MC
```

You can also do this on Windows if you leave out the `./` part.

Auto-start

As an alternative, if you would want later to set up auto-start of the Management Console as described in Starting Servers Automatically, you may select to do that already now instead of starting Management Console by hand.

Once the Management Console has been started (no matter how you do it), you must open its GUI in a browser. On Windows, you can do that with the aid of the "Management Console" item in the Start menu. On all platforms, you can open a browser and go to `http://localhost:50080/`. Accept the license terms and enter your license information including your license keys.

Design Studio

If Design Studio is not already started, do as follows:

Windows

Use the **Design Studio** item in the Start menu.

Linux

Start Design Studio from the command line by launching the DesignStudio program in the bin directory under the installation (see Important Folders in Kapow Katalyst) as follows:

```
$. /DesignStudio
```

You can also do this on Windows if you leave out the `./` part.

Next, Design Studio needs a license and will therefore show a dialog that looks as follows:

You have three choices, depending on what information you have been provided from Kapow and/or your system administrator:

License Server

If your administrator has provided you with the URL to a central license server (Management Console) that administers the license for all Design Studio users, select License Server and simply enter the URL and credentials for that server. The license server must be running and have available Design Studio seats in order for you to use Design Studio.

Developer License

The developer license is a combined license key for both Management Console and Design Studio. You can either enter the license key in your local Management Console, or use the dialog in Design Studio. Whenever you start Design Studio, the Management Console will automatically start as well.

Trial License

The trial license works in the same way as the developer license, but is intended for short-time use for trial/demonstration purposes.

Normally, Design Studio will display the **Enter License Information** dialog box only the first time it is started. It will, however, display this dialog box again if the license server cannot be contacted, for example because it has been moved or is not running.

Kapow Katalyst Initial Configuration

After installing Kapow Katalyst, you should configure the installation to suit your needs. Some configuration is done using the Settings application as described in the RoboServer Configuration section of the Administrator Guide.

Important For correct display of data in the Kofax Insight Dashboard, make sure Java correctly sets the time according to your time zone on RoboServers and computers running Management Consoles. See the *Timezone Data Versions in the JRE Software* on the Oracle web site for the latest updates in time zones. If necessary, use the *Timezone Updater Tool* to update the time zone information.

Other configuration, relevant mostly for administrators of the Kapow Katalyst system, is done via Management Console, more specifically via the Servers tab, the Projects tab and the Options tab. It is especially important to set up the necessary RoboServers and clusters on the Servers tab. Refer to the Administrator Guide and Management Console help for information on Kapow Katalyst configuration.

Starting Servers Automatically

If your installation includes any server functionality, you probably want to configure it to start the necessary servers automatically. One case where this is not relevant is if you only need to run Design Studio while being connected to a Management Console that e.g. your IT department has set up.

When referring to "server functionality", we mean RoboServers and the Management Console (license server). In fact these two functionalities are provided by the same server program, RoboServer, depending on the arguments given to it when it is started.

The section Starting RoboServer contains a detailed description of the command-line arguments to the RoboServer program and we will not repeat them here. Suffice to say that the `-service` argument is what enables the RoboServer program to execute robots. Similarly, the `-MC` argument is what enables the Management Console functionality (see also Starting the Management Console).

With this in mind, we will turn to the details on how to make RoboServer start automatically (whatever the arguments). This is quite different on Windows and Linux.

Starting Servers on Windows

To make RoboServer start automatically on Windows, you need to add it as a Windows service. We will show how to add and remove Windows services using the `ServiceInstaller.exe` program that is included in the Kapow Katalyst installation.

Adding Windows Services

To run RoboServer as a service you need to install it first using the ServiceInstaller.exe program. The following is a general example outlining the command-line arguments to this program (although displayed on multiple lines here, this is a one-line command):

```
ServiceInstaller.exe -i RoboServer.conf wrapper.nts.service.account=Account
wrapper.nts.service.password.prompt=true wrapper.nts.service.name=Service-
name wrapper.nts.service.starttype=Start-method wrapper.syslog.loglevel=INFO
wrapper.app.parameter.1="First-Argument" wrapper.app.parameter.2="Second-argument"
```

wrapper.nts.service.account

The account of the user that has to run RoboServer. Kapow Katalyst stores configuration in the user's directory and it is important to choose a user that has the correct configuration.

If RoboServer has to run as a domain user you need to enter the account in the form domain\account

If RoboServer has to run as a regular user you need to enter the account in the form .\account

wrapper.nts.service.password.prompt

The value true will prompt the user for the password for the account. If you prefer to enter the password on the command line, you must instead use wrapper.nts.service.password=whatever-the-password-is.

wrapper.nts.service.name

The name of the service to install. The name of the service can not contain spaces.

wrapper.nts.service.starttype

AUTO_START if the service should be started automatically when the system is rebooted.

DELAY_START if the service should be started after a short delay. (Not supported on Windows XP and 2003).

DEMAND_START if you want to start the service manually.

wrapper.syslog.loglevel

Redirect the console output from RoboServer to the event log.

wrapper.app.parameter.*

The arguments for RoboServer. You can enter as few or as many as needed.

When the service is installed the user will be granted the "log on as a service" rights. If the service fails to start, check that the right is granted by opening gpedit.msc and (on Windows 7) navigate to Computer Configuration > Windows Settings > Security Settings > Local Policies > User Rights Assignment > Log on as a service and add the user.

Removing Windows Services

To uninstall a service you can run

```
ServiceInstaller.exe -r RoboServer.conf wrapper.nts.service.name=Service-name
```

wrapper.nts.service.name

The name of the service to remove.

Example:

This example removes a service named RoboServer9.6.2

```
ServiceInstaller.exe -r RoboServer.conf wrapper.ntservice.name="RoboServer9.6.2"
```

Starting Servers on Linux

The simplest way to make RoboServer start automatically on Linux is to use crontab. Use the following command to create or edit the list of scheduled jobs in Linux for the desired user (it may be easiest to do this either as root or as that particular user):

```
crontab -u someUser -e
```

To the list of scheduled jobs add for example:

```
@reboot $HOME/KapowKatalyst_9.6.2/bin/RoboServer -p 50000
```

or

```
@reboot $HOME/KapowKatalyst_9.6.2/bin/RoboServer -p 50000 -MC
```

This way the RoboServer program will be started with the indicated command-line arguments upon reboot. Note that you must identify the bin directory under the actual installation folder.

Upgrading Kapow

In the interests of stability, different versions of Kapow may be installed side by side on the same computer without interfering with each other. This means that you can install a newer version and get acquainted with it while still doing your daily work with the older version. While doing so, you will probably want to open the older version's default robot project also in the Design Studio of the newer version. One thing to be aware of here is that if you save a robot with the newer version, you will not be able to open it using the older version of Design Studio. Thus it may be prudent instead to make a copy of the project and use the copy while you learn the new version of Design Studio.

Important If you use the API in your Kapow environment, the Java and .NET files must be updated when upgrading your copy of Kapow. The new API files are located in the API folder of your Kapow installation folder, for example, `C:\Program Files\Kapow Katalyst 9.6.0 x64\API`.

Below we will list the most important differences between Kapow versions that you should take into account when upgrading.

9.6

Read [Customers Upgrading to 9.6](#) topic if you are upgrading to Kapow version 9.6.

9.5

Read [Customers Upgrading to 9.5](#) topic if you are upgrading to Kapow version 9.5.

9.3

Version 9.3 of Kapow is fully backward compatible with 9.2 and your current robots will run with the Classic Browser Engine as in 9.2. For more information see [Existing customers upgrading to 9.3](#).

9.2

Version 9.2 of Kapow introduces new capabilities for extracting data from Microsoft Excel™ files. You will find advice on how to use these instead of the old ones in the section on [New Excel Capabilities in 9.2](#).

9.1

Version 9.1 of Kapow no longer includes the RoboRunner application. You will find advice on how to use Management Console instead in the section on [RoboRunner Discontinued in 9.1](#).

8.2

Version 8.2 of Kapow introduced some minor changes that in rare cases may affect the execution of older robots. Details on these and other changes are found in the section on [Changes in 8.2](#).

8.1

Version 8.1 of Kapow replaced the concepts of "models" and "objects" with the concepts "types" and "variables". Details on this major change are found in the section on [New Variables and Type System in 8.1](#).

8.0

Version 8.0 of Kapow introduced new ways to display a robot's control flow and error handling. This affects all robots, even those built with previous versions, as discussed in [New Control Flow and Error Handling in 8.0](#).

7.2

Version 7.2 of Kapow introduced a new storage system that is much easier to use. Users upgrading from the previous versions can switch to the new storage system at their own pace; this is discussed in detail in [New Storage System in 7.2](#).

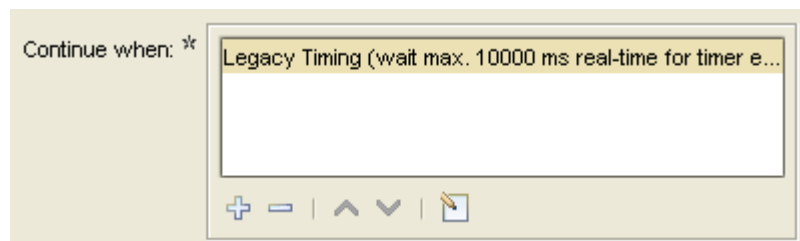
7.1

Version 7.1 of Kapow introduced a change in the location of certain files to comply with the installation rules of Microsoft Vista and Windows 7. Starting from this release, the files relevant to Kapow are divided into 3 folders: The Installation Folder, the Project Folder and the Application Data Folder. These changes mean that you no longer need to be a Windows power user (have write access to the "Program Files" folder) in order to run Design Studio.

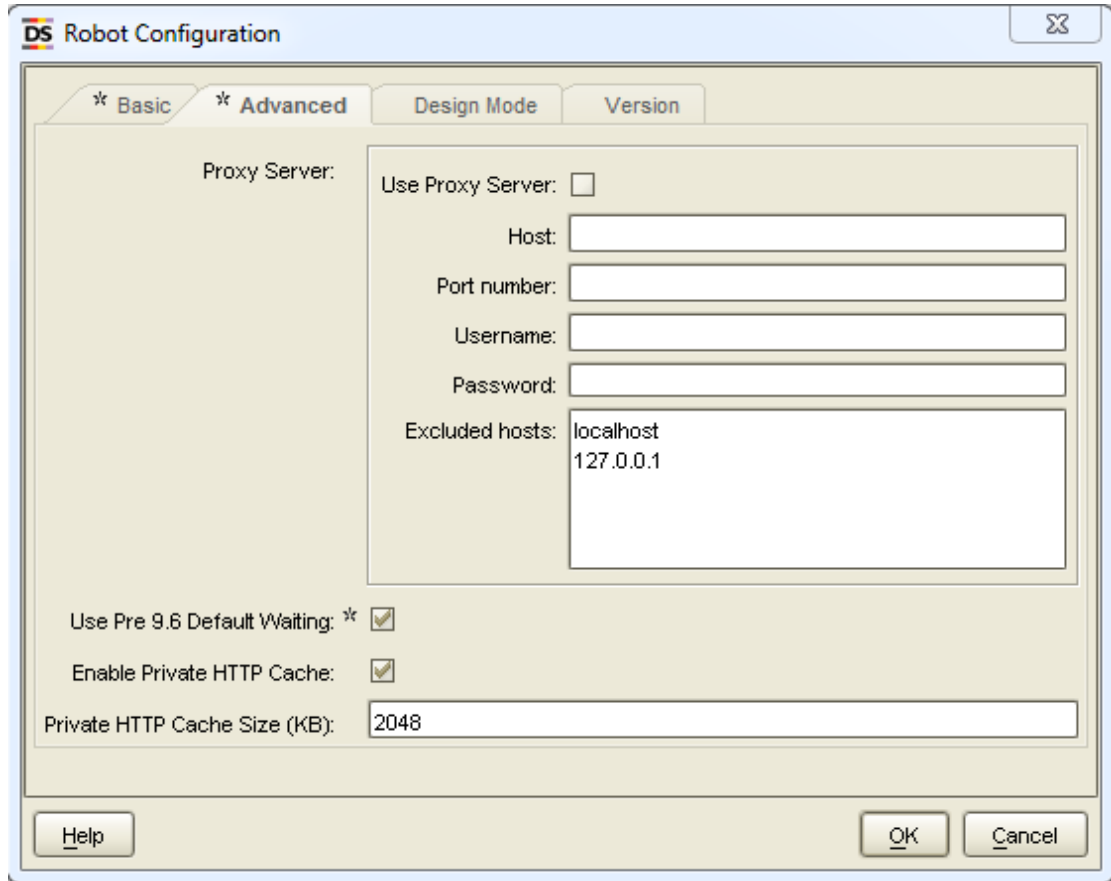
Customers Upgrading to 9.6

Upgrading Existing Robots to 9.6

A legacy wait criteria is added to Robots from previous versions (9.3, 9.4, 9.5) using the "Max. Wait for Timer Events" and "Wait Real-Time for Timer Events" for backward compatibility and ensuring the same behavior as in 9.5. This cannot be edited, but it can be removed and replaced with new criteria to improve performance if possible.



To switch to the new default settings when updating an existing robot to 9.6, go to the **File > Configure Robot** dialog box, click the **Advanced** tab and clear the **Use Pre 9.6 Default Waiting** check box.



Customers Upgrading to 9.5

Log Database

The log database format has changed in version 9.5, therefore if you want to use your 9.4 or older log database with Kapow 9.5, you must drop tables in your current database before using it with version 9.5. Follow the procedure to update the tables in your database:

- 1 Start the previous version Management Console and create a backup.
- 2 Stop your Management Console.
- 3 Backup the old logdb, if needed.
- 4 Drop the tables of the old logdb.
- 5 Start the Management Console version 9.5 and import the backup.

Customers Upgrading From 9.2

Kapow version 9.3 and 9.4 are fully backward compatible with 9.2 and your current robots will run with the Classic Browser Engine as in 9.2, there is no need to migrate existing working robots to the new Default browser. We currently recommend building new robots using the Default browser, but in case the web site does not render in the Default browser, we recommend switching to Classic. This will particularly be the case for legacy web applications design for compatibility with previous versions of Internet Explorer.

Please note that the Default browser is using Chrome as the browser to emulate, while the Classic browser emulates Internet Explorer. This can cause Classic and Default to render pages differently and return a different Document Object Model (DOM), so when migrating from a Classic to a Default robot, the tag-finders may be different and require modification.

Before upgrading a production environment to a new version, we always recommend testing the new version. We will be happy to provide non-Production license keys enabling you to perform the necessary testing. For more information about data migration from earlier versions see [Data Migration Tutorial](#).

Data Migration Tutorial

In this tutorial we will perform a test migration of existing Kapow data from version 9.2 to version 9.3 of the Kapow solution. We strongly recommend you to create a test setup to perform a migration, before you work with your production environment to ensure that Robots, restored in a new version of Kapow, do not interfere with your production process and database.

Note If you are using a Kapow version prior to 9.2, there are other considerations that are outside of this tutorial. Please contact Kapow technical support for assistance in migrating from version prior to 9.2.

Kapow version 9.2 and later use the new Kapow compute units licensing model as well as our centralized application settings, which are handled through the Management Console UI.

Most of the migration can be performed using the **Admin > Backup** tabs in the Management Console. In this tutorial we use MySQL as a data source for the project.

Any settings made in the RoboServer Settings application must be reset using the same application in the new version of the Kapow product.

Important For all the steps in this tutorial, including making changes in the RoboServer Settings application, we strongly recommend to use one and the same account that will be used later to run the new version of the Kapow application as a service on Windows or as a process on Linux.

Perform the steps of the following procedure to migrate your Kapow application data.

- 1 Start the Management Console version 9.2, open it in the browser, and perform a backup of your entire inventory. Later you will restore schedules, robots, types, etc. in the new version of Kapow.
 - a In the Management Console, go to the **Admin > Backup** tab.
 - b Click **Create Backup** under **Full Backup/Restore** and confirm your action.
This action will create a .zip backup file with all the configuration settings stored by the Management Console.

Note Kapow stores its data in the application data folder located under a Windows user folder of the user who is currently logged in the system. Because Kapow stores some of its important information in the data folder and because this information is relevant to the user account, we recommend you to include Kapow application data folder to your normal file backup procedure. For more information about Kapow folders, see the

[Important Folders in Kapow Katalyst](#)

- 2 Start your new version of the Management Console from the command line window. You should start the Management Console without starting the RoboServer by using the following command in the bin subfolder of the Kapow installation folder:

```
RoboServer -MC
```

The command window will show you the Management Console connector TCP port. The default port number is 50080. If you changed the TCP port using the RoboServer Settings application, the number you specified should appear in the command line window when the Management Console is being started.

Important Use the same account to start the command line interface and run the Management Console to ensure Kapow service has access to all the necessary files and folders.

- 3 Start the Management Console in the browser. If you use the same browser to switch between different versions of the Management Console, you might get the Unable to start application error with the The response could not be deserialized message. To avoid this shared browser cache issue, either use different browsers (for example, Internet Explorer and Chrome) for different versions of Kapow Management Console or you can try to just click OK and reload the page.

Note Once the Management Console is loaded, look for the version of the Console you are working with in the left top corner to make sure you working with the version 9.3 of the product.

- 4 Go to **Admin > Backup** tab and click **Restore Backup**. Specify the path to the zip backup file you created in your previous version of Kapow and click **Restore**. Once the restore process is complete click **Close** to close the **Restore Backup** dialog box.
- 5 Perform the following in the Management Console:
 - a On the **Admin > Clusters** tab, check that any listed RoboServers are not started. This action ensures that the new version of the Management Console does not connect to any of the previous version robots that you used when creating a backup in your previous version of the Management Console.
 - b On the **Admin > Clusters** tab, right click the cluster name and select **Cluster Settings....** Specify a database different from the one that was set in the previous version. If you made a backup from your production environment, this step ensures that your new robots would not interfere with your production database. You can choose to use the same database, but change the schema. It is strongly recommended to keep the name of the database without changing, because the imported robots will be looking for this label when running. Click OK and select to stop all robots and apply settings.
 - c On the **Admin > Settings** tab, click **RoboServer Log Database** and change the schema to a new clean version. Test the connection and save the settings.

Note If the specified logging database is empty, the application creates necessary tables when it connects to the database.

- d Check the Schedules tab and disable any scheduled runs that you do not want to be running.
- 6 On the **Admin > Clusters** tab, click **Add Server** to add a new server and assign this server to the production cluster. In this tutorial we will add a server that runs on the same machine where we run the new version of the Kapow Management Console. You can also specify a port number different from those you used for RoboServers in your previous version, for example, 50001.
The server you added appears offline in the Cluster/Server list until we start it with the specified port.
- 7 Open the command line window where you started the Management Console and stop the process (Ctrl C).
- 8 Start the Management Console and the RoboServer with the port you specified using the following command:

```
RoboServer -MC -p 50001
```

Once the RoboServer is started, you can see the version of the server listed in the Cluster/Server list on the Clusters tab.
- 9 Create necessary tables in your new database. You can use database tools to populate tables from your production database, or you can use the Design Studio to create necessary tables. You must specify the same database for the cluster in the Management Console and for the project in Design Studio. To create the tables using the Design Studio, download and save necessary Types and Robots from the Management Console to your new project folder, open the project in Design Studio, specify all the database settings, and allow the Studio to create necessary tables in the database. To check that the tables are created, go to the Data tab in the Management Console and look for the tables in the selected database.
- 10 If you imported any Kapplets from your previous version, go to Kapplets>Repository in the Management Console and enable the Kapplets you want to use. Once the Kapplets are enabled, you can open the KappZone and check whether the Kapplets can be opened and run.

Important After you complete the migration, perform a thorough examination of your existing processes in the new version of Kapow Katalyst to confirm that everything is functioning as expected. For any questions during migration and verification testing, contact Kapow technical support by sending an email to support@kapowsoftware.com

New Excel Capabilities in 9.2

Since the way Excel is handled in 9.2 fundamentally different from the way that Excel was handled in previous versions of Design Studio it is important to know that there is no simple way to upgrade robot from earlier version to 9.2 when these involves Excel. It may involve a considerable amount of rewriting. This is because in previous versions Excel was transformed into HTML during loading and subsequent extraction etc. was then done in the same as for HTML, but in 9.2 Excel is loaded into the new Spreadsheet Page View and dedicated steps are used for data extraction from this.

The Convert Excel to HTML Legacy Option

Old robots still works since there is a legacy option called Convert Excel to HTML that allows this and old robots have this option turned on by default. If you want to upgrade a robot to use the new Excel capabilities you have to turn this option off. You do this under **Robot Configuration** by clicking the **Configure Options** button and on the **Legacy** tab of the **Configure Option** dialog box you clear the option. But changing this would, unless you do not do any extraction or looping that works on the Excel, break your robot. You will have to go through your robot and replace some of the step with the new dedicated Excel steps where ever the robot work on the data from the loaded Excel document. So unless you have compelling reasons for upgrading your old robots to use the new Excel capabilities you should probably not do it. This does not mean that you cannot upgrade your robot to use other 9.2 feature, just that you should leave the legacy option turned on.

Note The Convert Excel to HTML option is always set in your old robot even if it did not involve any **Extract from Excel** steps or Excel in any other way. It is important to remember this in case you load an old robot into 9.2 and use this as your starting point for building a new robot, because if you do not turn off the Convert Excel to HTML option then you will not be able to take advantage of the new Excel capabilities.

The Extract from Excel Step Action is Deprecated

The Extract from Excel step action has been deprecated and cannot be used in new robots instead you should use a **Create Page** step action for this.

RoboRunner Discontinued in 9.1

As previously announced, the RoboRunner application is no longer supported. This guide will show you what to do instead.

The task of scheduling robots at given intervals is now handled by the Management Console, which is a web based application. We will look at each of the different RoboRunner execution scenarios, and how these tasks are done using the Management Console. Before you proceed any further we recommend that you read the Management Console chapter in online help, as we will be referring to it in great detail later.

Although the robots are scheduled by the Management Console they are in fact executed on a RoboServer. You can read more about the RoboServer in the Administrator's Guide. The RoboServer can be monitored using the Management Console dashboard.

Scheduling concepts

RoboRunner had two ways of executing robots:

The **-file** option

If your robots were scheduled individually using RoboRunner, you will have to create a separate schedule for each robot to have it run at the desired time, as described here. Scheduling robots this way has an added benefit, as you can now parametrize your robots using input objects.

If you have multiple robots that need to run at the same time, you can create a schedule by selecting the robots on the **Robots Tab** in Management Console, right clicking and selecting **Create Schedule** from the context menu.

Bulk deploying to Management Console

Once you have structured your robots, you are ready to deploy them to Management Console. The deployment procedure is different for different versions of Kapow.

If you are deploying your robots in Kapow version 9.5 and later, just connect to a Management Console from your copy of Design Studio and upload the project to the selected Management Console. For more information, see *Management Consoles* and *Working with Shared Projects* topics in the *Design Studio* section of the online help.

If you are deploying your robots in Kapow version prior to 9.5, create a Project Export from within Design Studio. To create a Project Export, perform the following steps:

- 1 Right click the project folder in the project view, and select **Create Project Export** on the context menu.
- 2 Select a location where you want to store the project export, and click OK.
- 3 If any of your robots were developed before version 8.1, expand any `.model` files you may still have.

You can read more about types and models in *New Variables and Type System in 8.1*. Also, names of robots in a project must be unique even if they are located in different sub-folders. A dialog box will inform you if there are any name conflicts.

Once you have created the project Export, open the Management Console and go to the **Options Tab (Admin > Options)** to perform the Project Import.

Robots Schedule

Once your robots are uploaded, you can create schedules to execute your robots.

If your robots use the old database storage system (Database Storage/Message/Info Environment), you will have to configure options on the **Legacy** tab of the **New Schedule Dialog**. However, we recommend that you rewrite these robots to use the new storage paradigm introduced in 7.2. You can read about the new storage system at *New Storage System in 7.2*.

The new storage system includes a new and improved database logging system, which is integrated into the Management Console, and allows you to 'jump' directly from a schedule to the errors generated by the robots. This will work in parallel with the legacy Message Environments, but again we recommend that you rewrite your robots, since the Storage/Message Environments will be removed from the product at some point in the future.

Changes in 8.2

This section is meant only for users of older versions of Kapow Katalyst. It will detail changes done in version 8.2, some of which may affect the execution of older robots:

- When all branches of a Try step fail, the situation may be logged or reported back to the caller of the robot as an API exception, depending on how the Try step is configured (this is described in detail in *How to Handle Errors*). When running the robot on RoboServer in versions prior to 8.2, all the details on what caused each branch to fail were logged or reported. However, it was not stated that all branches of the Try step failed, and the name or location of the Try step was not given either (this information was available only in the debugger).

Starting with version 8.2, it will be reported or logged as a separate error that "All alternatives of the Try step failed", including a reference to the Try step. This error will be reported or logged prior to the details about the failed branches. Note that this change may affect some API clients and will affect the contents of the robot property Robot.executionErrors which can be used in expressions in the robot.

- If you are using database logging you will have to updated one of the log tables. The ROBOT_MESSAGE in the log database now has an additional column named PROJECTNAME. Before you can use database logging you must update the ROBOT_MESSAGE table. There are two option, either delete the ROBOT_MESSAGE table (you will loose all errors/ messages from previous runs), the table will be recreated when RoboServer (or Management Console) is restarted. If you wish to keep history from previous runs, or the credentials used by RoboServer doesn't have the ALTER TABLE privilege you should have your DBA add the PROJECTNAME column. The type of the column should be VARCHAR(255) (NVARCHAR2 on Oracle, NVARCHAR on Sybase and Microsoft SQL server).

New Variables and Type System in 8.1

This section is meant only for users of older versions of Kapow Katalyst. It will further the understanding of the new variables and type system in version 8.1.

As of version 8.1, the terms "object" and "model" are no longer used. A robot can now contain a number of variables, and each variable has a type. This new terminology is meant to make things more clear, as the term "object" was previously used rather loosely. Overall, the following changes have been made:

- The terminology is changed, so that what was in the model editor called objects is now called types, and what was in the robot called objects is now called variables. The Model Editor is therefore now called the Type Editor
- Types and variables do not work in the same way as objects did. A type should only be viewed as a blueprint for a variable, that is, variables can be created from a certain type. The old terminology indicated that **objects** were in fact created in the model editor, and could then be added to the robot, if necessary. This is no longer the case. Therefore, it is an important difference to understand: types do NOT define variables, but only blueprints, and to make use of a type in a robot, a variable OF that type must be created.
- A type file, as opposed to an old model file, only defines a single type. Previously, model files could define any number of objects, making it difficult to maintain an overview.
- It is possible to create more than one variable of a certain type. This is opposed to previously, where an object defined in a model file could only be added to the robot once.
- A number of simple types exist that can be used for storing a single value. The purpose of these, among other things, is to replace the ScratchPad object. Also, in many cases, they make creating a robot easier as they can be used instead of having to create types simply for storing temporary data. To be able to distinguish, the types created by the type editor are referred to as complex types. Only variables of complex types can be used as input to the robot, or returned as output. Variables of simple types should thus be considered internal robot variables.
- A complex type is not defined as being used for either input or output. Instead, it is possible on a variable to set whether it should be used as input. The output term is no longer used; all variables can be output (except variables of simple types which, as mentioned above, cannot be used as either input or output).

- The old model files can still be used, but can no longer be edited. The contents of the model files is now interpreted as types, and not as objects. This means that the types defined in a model file can be used to create variables - just as types defined in the new type files. In model files, objects were defined as being either input or output. This information is no longer needed, as input is defined on variables. Therefore, it is simply discarded when loading model files. As an example of when to be aware of this is when creating a variable from a type defined in an old model file. If the object from which the type is derived was previously an input object, it is important to note that the variable created from the corresponding type will NOT automatically be set to input. This must be done manually.
- Model files can be converted to type files. This creates a single type file for each type (object) defined in the model files, and these can then be edited if necessary. A Model is converted into Types by right clicking the model in the project view (in Design Studio) and selecting **Expand Domain Model File** on the context menu.
- The old concept of global variables has been changed. Now, any variable can be set as global. Being able to make any kind of variable global provides a higher degree of freedom in how to structure a robot. When opening a robot using the old global variables, these are converted into the new kind of global variables of the simple type Short Text.

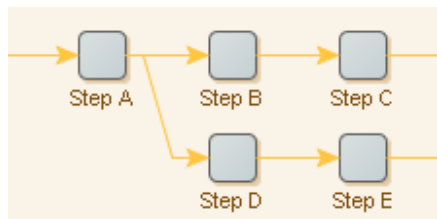
New Control Flow and Error Handling in 8.0

Version 8.0 of Kapow introduces a new way to display a robot's control flow and contains extensive changes to the way errors are handled. This affects all robots, even those built with previous versions, meaning that although they still work in the same way, these robots will look differently when opened in version 8.0 or newer. For the benefit of users of the earlier versions of Kapow, this chapter will give examples of the differences.

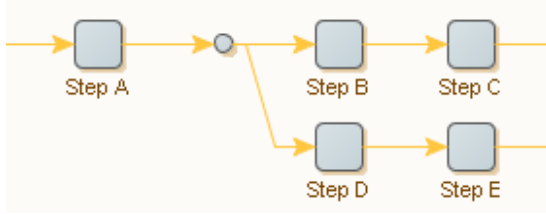
Branch Points and Try Steps

The most visible change is that ordinary steps now can have only one outgoing connection. If the step has more than one step following it, an explicit branch point (a small circle) or a Try step is inserted after it. These new steps are the only ones that can have more than one connection going out from them.

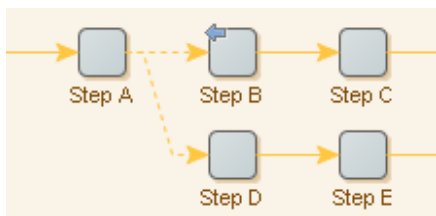
This also means that the "Branching Mode" property has disappeared from the step configuration. If a step previously had the "All Branches" branching mode, it will now be followed by a branch point as seen in the following example. First a snippet of a robot as it looked in version 7.2:



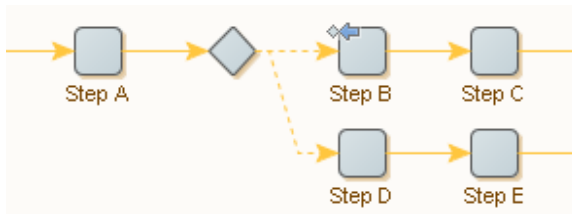
In version 8.0, the same robot now looks as follows:



Of course a branch point is only inserted after a step if it has more than one step following it. If the step previously had the Until Successful Branch branching mode, it will now be followed by a **Try** step rather than a branch point. This step is a core component of the new way to handle errors and is described in the Error Handling topic; at this point we will merely show an example of what it looks like. Consider the following part of a version 7.2 robot:



When you open this robot in version 8.0, it will look this:



End Steps

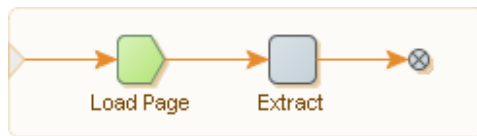
The introduction of End steps is a small but important change. An End step does nothing (in particular it does not end robot execution), but it provides an explicit and automatically inserted "last step" on each and every branch in the robot. This means that if you click on an End step (in Design Mode), all of the branch will be executed. Previously, Do Nothing steps were often added manually to serve the same purpose. This is no longer necessary.

A single explicit "begin robot" marker (a right-pointing triangle) is also displayed in each robot. It is provided merely for balance, and because it makes a robot a bit more similar to a group step (another new feature).

The following complete robot shows both End steps and the marker at the beginning of the robot. In version 7.2, the robot looked like this:

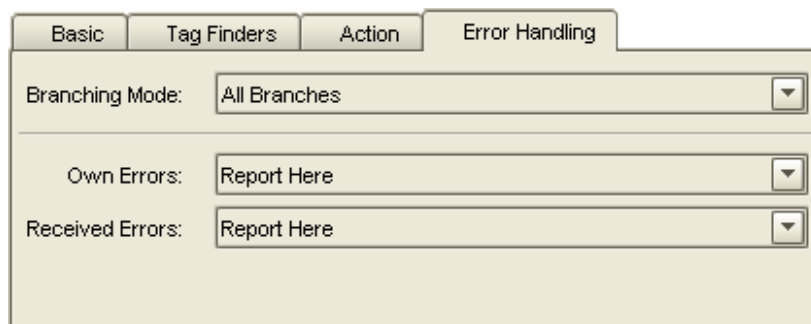


And when opened in version 8.0, the same robot looks like this:

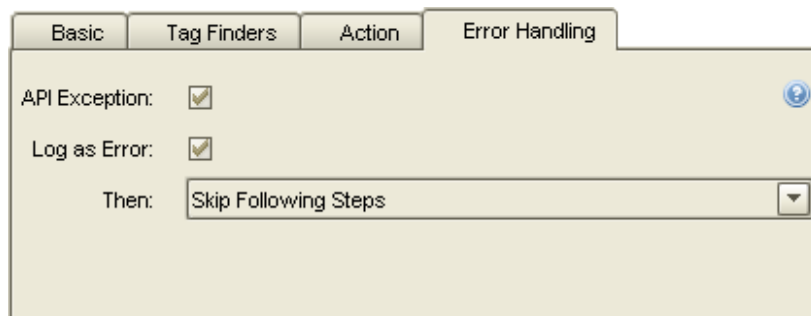


Error Handling

The area of error handling in robots has been thoroughly revised. The section ... explains this in detail; in the following we will merely indicate how the older way of error handling translates into the new one. In version 7.2, error handling was configured by way of the following properties:



All of this has been replaced by the following set of properties, which are explained below:



Errors May Be Reported or Logged No Matter How They Are Handled

It used to be that the step property "Own Errors" specified both whether to "report" an error and what effect the error should have on further execution of the robot (error handling). These aspects have now been separated and can be configured independently. Instead of "Own Errors" we now have three properties: "API Exception", "Log as Error" and "Then".




No matter how the error is handled (even if it is ignored), the incident can be reported back to the caller of the robot (this is named an "API Exception"). In addition and independently of this, the error can be logged ("Log as Error"). Both API exceptions and logging can be selected or deselected for each step independently.

The error handling aspect of the former "Own Errors" property is now taken care of by the "Then" property.

Errors Are Handled Where They Occur

Previously, one very important way to handle an error was to specify that it should be sent backwards (to a step further to the left in the robot) for handling. The configuration of Received Errors and Branching Mode on the (possibly many) traversed steps would then decide how the error would ultimately be handled. It might end up being reported (causing an API exception); affect the execution of the robot; or be ignored, and it was impossible to know without considering the configuration of all traversed steps.

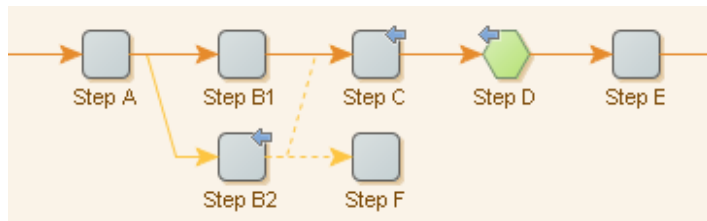
This whole notion of sending an error backwards has now been replaced by a more direct approach, where you always specify on the step where the error occurs what to do with it, without depending on the configuration of other steps. Thus the Received Errors property has been removed.

When an old robot is opened, an analysis is done for each step that uses Send Backwards in order to determine how to assign the Then property. Most often, the result is that instead of Send Backwards (displayed as  on the step), a step will now use either Try Next Alternative () or possibly Skip Following Steps () in combination with API Exception and Log as Error.

Send Backwards is still available as a way to handle errors, but is deprecated (meaning you cannot select it when you edit the Then property). The only reason it is still available is that it is necessary in order to execute some robots in exactly the same way as in previous versions of Kapow Katalyst. It is needed only by a small number of robots. When examining these robots, you should think of all steps as having Received Errors set to Send Backwards.

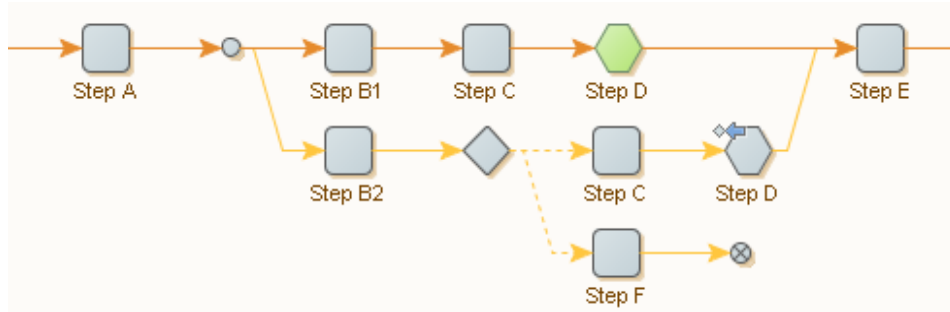
Small Changes in Robot Structure

In rare cases where branches merge, the result of the analysis described above may be that the step's Then property should be configured in different ways depending on the path followed to the step. In such cases, the step (and often small parts of the robot) will be duplicated to de-merge branches in a way so this dependency on the path disappears. Thus you may see some surprising expansion of the robot. The following robot is an example (it may seem contrived, but robots that exhibit this behavior are odd). The robot looked like this in version 7.2:



When step D is executed, it may report an error. If the step is reached on an execution path that goes via step B1, the error will be sent back to B1 and will be reported there. However, if step D is reached via step B2, it will not be reported. Rather, it will cause execution to proceed at step F.

When opened in version 8.0, the robot is automatically modified as follows:



We see that step D has been duplicated and that the two copies have been configured differently. The one associated with step B1 (in green) has API Exception and Log as Error turned on and uses Skip Following Steps, while the one for B2 does no logging, but uses Try Next Alternative. Step C has been duplicated also, in order to make it possible to connect the steps correctly.

Small Changes in the Order of Error Reports

It should be noted that in earlier Kapow versions, Send Backwards made it possible to delay reporting an error as an API exception. In version 8.0, however, if an error is reported, it is reported as soon as it occurs. For some robots, this means that errors are reported back to the caller in a different order than previously, and they may be grouped into API exceptions in a different way, too.

New Storage System in 7.2

With version 7.2 of Kapow storing data in databases has become easier than with previous versions. The new storage mechanism lowers the learning curve and eliminates tedious tasks, allowing you to spend more time writing robots.

This guide describes the differences between the old and new storage systems, and the upgrade path required to move to from the old to the new storage system.

The Old Storage System

Since the first release of Kapow it has been possible to store data in databases. Although the system has proven its worth, a number of problems have become visible over time.

Problems with the old storage system

- Storing data in a database required a special DatabaseOutputObject.
- The DatabaseOutputObject had a `refundKey` field; few understood the importance of the `refundKey` at the time the object was constructed, and once data had been stored, changing the `refundKey` was a cumbersome process.
- You had to register each robot with the RoboManager database, so the robot would have a `robotId`.
- At run-time you had to provide 3 environments, a DatabaseStorageEnvironment, a DatabaseRobotInfoEnvironment and a DatabaseMessageEnvironment.
- The Refind Object action was confusing, and error prone to use.

These issues introduced a number of pitfalls, which even experienced users fell into. Most common is probably the duplicate `robotId`, where an already registered robot was used as a

template, but the copy was never re-registered with RoboManager, and suddenly two robots had the same RobotId.

The New Storage System

The new storage system, introduced with version 7.2 of Kapow Katalyst, intends to fix the shortcomings of the old storage system.

Benefits of the new system

- No dedicated OutputObject for databases, just use regular output objects.
- No registration of robots.
- Data is stored using the Store in Database step action.
- No environments, logging is optional and configured using the Settings application.
- No RoboManager, the log is viewable through a web interface.
- New Find in Databases, and Delete in Database step actions.

The new system has completely removed the use of RoboManager and environments, which greatly simplifies the process of running a robot that collects data into a database.

Instead of the `refindKey`, you now have an `ObjectKey`, but in addition to configuring it when you build the model, it can also be specified inside the robot (also no fields are selected for the `ObjectKey` by default, forcing you to consider which fields define object identity).

When an object is stored using Store in Database, 7 additional (household) fields are stored. These are similar to the 6 hidden fields contained in the old DatabaseOutputObject, except for the following changes:

- `ObjectKey` replaces `refindKey`.
- `ExecutionId` replaces `RobotRunId` (allows you to correlate with API executions).
- `RobotName` replaces `RobotId` (so make sure the robot names are unique).
- `LastUpdated` is added, this allows you to see the last time this record was updated (similar to the functionality added by the old ChangeDate plug-in).

Legacy Options

After a fresh install of 7.2 the old storage system is completely hidden, this is done to prevent confusion for new users. If you need to build old Database Output Objects, or run a robot with environments in the Debugger, you have to enable Legacy Properties in Settings; this is done on the Legacy tab.

Backward Compatibility

Since the two storage systems required different table layouts, they are not compatible. However, in most scenarios the two systems can coexist without problems, allowing you to migrate your robots over time. You should not mix the usage of the new and the old system within a single robot.

In most cases you can execute robots using both storage systems on the same RoboServer, as long as you don't provide a DatabaseRobotInfo Environment to robots which use the new storage system. This means that if you use a deployment descriptor or API code to configure the environments, you will have to make a distinction in your code between robots using the new and the old system. Alternatively you can temporarily give new robots a RobotId

allowing them to be run with the RobotInfo/Message environments, and then remove the environments once all robots have been migrated.

If a new robot is executed with a Database Storage Environment, it will be ignored, as long as the robot doesn't contain any old Database Output Objects or Return Object steps.

If you execute a robot with both a Database message environment and database logging enabled for RoboServer, you will get two different logs: One you access through RoboManager, and another one you access via the Log View in the Management Console.

Migration

Migrating from the old to the new storage system usually requires four distinct migration steps. Migration of the models, the robots, the data, and any SQL scripts interacting with the data tables.

Migration of Models

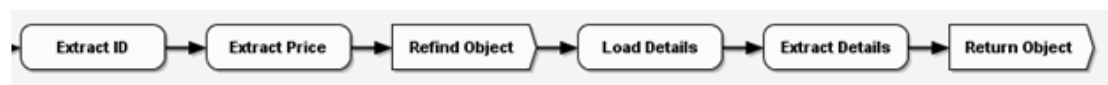
Change the model type from DatabaseOutputObject (Legacy) to OutputObject and remove the 6 hidden fields (robotId, robotRunId, refindKey, firstExtractionDate, latestExtractionDate, extractedInLatestRun). Consider which fields should be part of the ObjectKey.

Migration of Robots

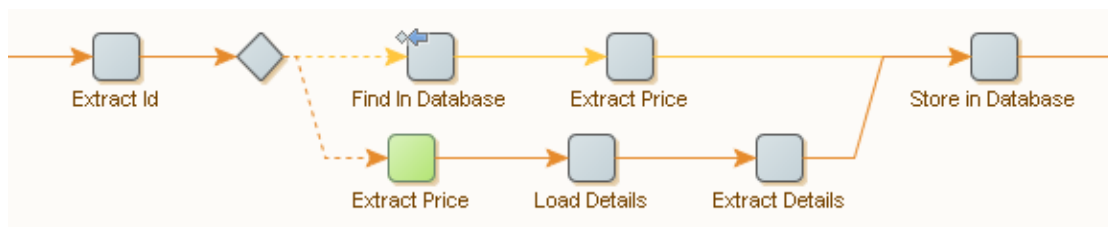
The following steps must be completed to migrate your robots.

- Replace Return Object with Store in Database, and configure it appropriately.
- Replace Refind Object with Find in Database. Since Refind Object will update the object and stop if the object was found, the structure of the robot has to be changed a little.

Here is a classic scenario using Refind Object. The object Id is extracted (the only attribute used for refindKey in this example), then the price is extracted (not part of refindKey), then the object is refound. If the object exists, the price is updated (and the rest of the branch is skipped), if the object doesn't exist, the details page is loaded and the details are extracted and the object is then returned.



This should be converted to:



First we extract the id (Object Key attribute) then we use Find in Database to check if the object already exists. If the object exists the current record is loaded into our output object, and we now extract the price on-top of the previous value, and then we store the object again. If the object was not found by Find in Database an error is generated, and "Try Next

Alternative" error handling sends the execution into the lower branch where we extract the price, load and extract the details, and then store the object.

The new method may require an extra Extract Price step, but it is much easier to see what happens, and there is no refind-object-magic.

If your robot has any Query Database or Execute SQL steps which query the storage tables, these may need to be updated as well.

Migrating Existing Data and Tables

If you have small amounts of data, or you collect all the data every time your robots run, the easiest is simply to delete (drop) the existing tables, and create new tables using the updated model objects.

If you have large amounts of data, or need the historic data from previous runs, you have to convert your existing data tables into the new table layout. It is possible to use tools from your database vendor to add/remove and rename the fields, but it is not recommended. The best solution is to use a robot to load the old data and store it using the new format.

To do this, create a table from you updated output object; you will have to rename the original data table unless you have given the new version of the object a different name. Then create a robot which loads the existing data using the Query Database step action, and stores the objects using Store in Database.

If you require the data history, like first and last Extracted date, you can extract these from the old row, and use the Execute SQL step to update the object after it has been stored.

If you have exposed the refindKey outside of your Kapow environment, you probably need to track which ObjectKey maps to an existing refindKey, and then provide the outside world with a mapping table, allowing them to convert existing refindKeys into the new ObjectKeys. This is important since you can't always generate an ObjectKey identical to the previous refindKey, since the robotId was often part of the refindKey calculation.

Migrating SQL Scripts

It is not possible to give a generic rule for how to migrate external SQL scripts, since the possibilities and complexity of such scripts are virtually endless. Instead we will try to give some guidelines.

Moving production data out of the harvest database

It is not a good idea to run user queries against the harvest tables (the tables the robots collect into), instead you should have a separate set of tables for the production data. These tables will contain the refindKey/ObjectKey, but not necessarily any of the other household fields. The production tables usually should be updated each time a robot has executed successfully.

Previously, the rows to move was often identified by using the robotRunId column to find the most recent run, but this column no longer exists. Instead we now have a column named LastUpdated, which contains a timestamp of when this row was last updated. This way all you need to do is find all rows which have been updated since you last moved data out of the harvest tables. If you have multiple robots harvesting data into the same table, you now need to use the robotName and not the robotId to distinguish data from different robots.

Scripts against RoboManager tables

The new database log offers superior information compared to the old RoboManager logs, some of these (like logging of Input Objects) are for internal use, but the rest are viewable in log view of the Management Console. If you have scripts containing queries against the RobotRun or Message tables in RoboManager, these will need to be rewritten to match the new logging layout.

Most significantly the robotId and robotRunId have been replaced with RobotName and ExecutionId. The advent of the executionId allows you to trace a specific run or message to an execution created using the API, something which was not previously possible. The new ROBOT_RUN table also contains a number of extra statistics, such as queue time, execution time and total runtime.

Index

G

getting help, [6](#)

H

help for Kofax products, [6](#)

S

support site, [6](#)

