TRUSTED RUBIX™
Version 6

Installation and Quick Start Guide
Red Hat Enterprise Linux 6 SELinux Platform
Revision 6

RELATIONAL DATABASE MANAGEMENT SYSTEM
Infosystems Technology, Inc.
4 Professional Dr - Suite 118
Gaithersburg, MD 20879
TEL +1-202-412-0152
# Table of Contents

OVERVIEW .................................................................................................................. 1

INSTALLATION ........................................................................................................... 1

  OPERATING SYSTEM INSTALLATION ..................................................................... 2

  TRUSTED RUBIX 6.0 INSTALLATION ....................................................................... 3

    Trusted RUBIX Server Platform Installation ....................................................... 4

    Trusted RUBIX Client Platform Installation ....................................................... 5

CONFIGURING LOCAL COMMUNICATIONS ................................................................. 5

CONFIGURING REMOTE COMMUNICATIONS .............................................................. 6

CREATING A DEVELOPMENT USER .......................................................................... 7

EXECUTION AND USE ............................................................................................... 9

  ROLES AND ROLE TRANSITIONS .......................................................................... 9

    Operating System (OS) Roles ............................................................................... 10

    Trusted RUBIX Roles ......................................................................................... 12

STARTING THE TRUSTED RUBIX DISPATCHER ...................................................... 13

CREATING A DATABASE ............................................................................................ 13

PERFORMING CLIENT SQL OPERATIONS ............................................................... 14

CONFIGURING AND CREATING ODBC APPLICATIONS ....................................... 14

UNINSTALLING TRUSTED RUBIX ............................................................................. 15

OVERVIEW OF USER DOCUMENTATION ............................................................... 15

SUPPORT AND FURTHER INFORMATION .................................................................. 16
Overview

This document describes the installation procedures and basic operation of the Trusted RUBIX RDBMS software. It applies to installing Trusted RUBIX 6 on the Red Hat Enterprise Linux 6 (RHEL6) operating system on i386 and x86_64 architectures. The procedures and processes described here apply only for version 6 of the Trusted RUBIX software.

Installation

The installation must be performed on the RHEL6 SELinux operating system running on i386 and x86_64 architectures. Configuration of the operating system and installation of Trusted RUBIX must be performed as the Linux root user. Furthermore, the SELinux security policy should be set to permissive mode or the user must assume the unconfined_r role under the Targeted policy. It is recommended to use the Targeted policy, the root Linux user, and the unconfined_r role. For the Targeted policy the unconfined_r role is the default login role for Linux users.

It is also possible to simply place the SELinux policy into permissive mode and become the root user. In permissive mode the SELinux policy enforcement is disabled and any operation that would have been denied due to the SELinux policy is audited. The SELinux policy may be placed into permissive mode using the GUI SELinux Management tool. It may be started from the System -> Administration -> SELinux Management menu. Alternatively, you may issue the “setenforce Permissive” shell command. Permissive mode should only be used for development and configuration and not for deployment.

For Trusted RUBIX to operate properly, the SELinux policy on RHEL6 must either be operating with the Targeted policy or the MLS policy. The Targeted policy protects a limited number of OS functionalities while MLS policy protects all operating system functionalities. Trusted RUBIX enforces its SELinux behavior in the same way with either policy but the operating system will be easier to configure and use operating with the Targeted policy and the unconfined_r role.

The Targeted policy is the default policy configuration for RHEL6 and tends to be less problematic. It is strongly recommended to use RHEL6 / Trusted RUBIX with the SELinux Targeted policy until becoming very familiar with SELinux policy behavior. It is possible to switch between the Targeted and MLS policies using the SELinux Management tool; however, databases created under one policy will not function under the other policy.

The Trusted RUBIX SELinux Guide has more information about SELinux and its relationship to Trusted RUBIX.

The book SELinux by Example (ISBN: 0-13-196369-4) provides a good, but somewhat outdated, introduction to using and configuring SELinux.

For more recent information about SELinux see the SELinux Notebook at the following URL:

http://www.freetechbooks.com/the-selinux-notebook-the-foundations-t785.html

For more information about the SELinux policy on RHEL6 see the following URL:

https://access.redhat.com/knowledge/docs/Red_Hat_Enterprise_Linux/
General information about SELinux, including an active mailing list, may be found at:

http://www.nsa.gov/research/selinux/

Operating System Installation

During this step you will install the RHEL6 operating system on your hardware, apply all available system updates, and install any packages needed by Trusted RUBIX. Please ensure that you install the version of RHEL6 that corresponds to your version of Trusted RUBIX.

Install RHEL6 on your hardware as described by the RHEL6 installation instructions. If you intend to develop Trusted RUBIX software it is recommended that you select the “Software Development Workstation” configuration during the installation. After your installation has completed, login to the system using the user and password you created during the installation. By default, the Targeted Policy will be in Enforcing mode. Perform the following steps:

1. Open a command terminal from the Applications->System Tools->Terminal menu.
2. Become the Linux root user by issuing the `su -` command. Enter the root password when requested.
3. Verify your current Linux user and SELinux context by issuing the `id` command. You should observe output similar to the following:
   
   
   
   uid=0(root) gid=0(root) groups=0(root)
   context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023

4. Update the system by issuing the following command: `yum update`. Note that an Internet connection is required for this step.
5. If required, reboot the system, re-login, and repeat steps 1-3 from above.

If your platform has access to the Internet and the yum command will be used to install the Trusted RUBIX 6.0 packages then all dependent libraries will be automatically downloaded and installed. In this case you may proceed directly to the section titled “Trusted RUBIX 6.0 Installation.”

The yum install operation may be used to download and install a package, as shown below. Note that using the yum install operation will ensure that all dependent packages are downloaded and installed as well as the target package.

```
yum install PACKAGE_NAME
```

Information about a package, including its installation status, may be found using the following command:

```
yum info PACKAGE_NAME
```

An Internet connection is required to perform the yum operations and the command will automatically locate the proper package repository on the Internet. If you do not have a network connection on the host machine you will need to manually download the package files and all dependent packages on another machine and then transfer them to the host machine.

Installing an individual package file that resides on the host machine may be accomplished with the following command (from the directory containing the package file):

```
yum localinstall ./RPM_PACKAGE_FILE
```
Install all of the following packages and their dependencies as the Linux root user and unconfined_r role. Note that some of the packages need to be specifically for the i686 architecture. Depending on the configuration of your operating system installation, some of these packages may already be installed; if so, you will receive an appropriate message during the package installation operation.

- selinux-policy
- selinux-policy-targeted or selinux-policy-mls (or both)
- netlabel_tools
- policycoreutils-python
- policycoreutils-gui (recommended but not required)
- policycoreutils-newrole
- libselinux.i686
- mcstrans
- libcap.i686
- zlib.i686
- openssl.i686
- selinux-policy-devel (needed for policy development only)
- setools (needed for policy development only)
- setroubleshoot (needed for policy development only)
- unixODBC.i686 (only if platform will run ODBC clients)
- unixODBC-devel.i686 (only if platform will be used for ODBC client development)

To install all of these packages in one operation, issue the following command as the Linux root user and unconfined_r role:

```bash
yum -y install selinux-policy selinux-policy-targeted selinux-policy-mls netlabel_tools policycoreutils-gui \
 policycoreutils-newrole libselinux.i686 mcstrans libcap.i686 zlib.i686 openssl.i686 selinux-policy-devel setools \
 setroubleshoot unixODBC.i686 unixODBC-devel.i686
```

**Trusted RUBIX 6.0 Installation**

Ensure you are the Linux root user, unconfined_r role, and the Targeted policy is active. Place the Trusted RUBIX 6.0 installation disk into your RHEL6 computer. Note that a copy of this installation guide is included on the disk. Copy the following files to your computer (the x's will be replaced by your actual version numbers):

- rubixdbs-policy-base-mls-6.x.x-x.el6.ixxx.rpm (server machine only)
- rubixdbs-policy-base-targeted-6.x.x-x.el6.ixxx.rpm (server machine only)
- rubixdbs-policy-devel-mls-6.x.x-x.el6.ixxx.rpm (server machine only)
- rubixdbs-policy-devel-targeted-6.x.x-x.el6.ixxx.rpm (server machine only)
- rubixdbs-6.x.x-x.el6.ixxx.rpm (server machine only)
- rubixdbs-devel-6.x.x-x.el6.ixxx.rpm (server machine only)
- rubixdbs-doc-6.x.x-x.el6.ixxx.rpm (any machine for which you want the documentation installed)
- rubixdbs-odbc-6.x.x-x.el6.ixxx.rpm (client machines without the rubixdbs-devel package installed)

If you are upgrading from a previous version of Trusted RUBIX, it is recommended that you first
uninstall the old version. Your data files will not be removed and a backup of the rxconfig file will be automatically saved by the yum tool.

**TRUSTED RUBIX SERVER PLATFORM INSTALLATION**

The Trusted RUBIX package names for server platforms are `rubixdbms-policy-base-mls`, `rubixdbms-policy-base-targeted`, `rubixdbms-policy-devel-mls`, `rubixdbms-policy-devel-targeted`, `rubixdbms`, `rubixdbms-devel`, and `rubixdbms-doc`. All client-only platforms and any server platform which does not have the `rubixdbms-devel` package installed may obtain the ODBC client libraries from the `rubixdbms-odbc` package.

For each platform that will be a Trusted RUBIX server, install each package file in the order given using the following commands as the Linux root user and unconfined_r role (the x's will be replaced by your actual version numbers):

```bash

cd RPM_PACKAGE_FILE_DIR
Only if selinux-policy-mls is installed:
yum --y localinstall /rubixdbms-policy-base-mls-6.x.x-x.el6.ixxx.rpm
Only if selinux-policy-targeted installed:
yum --y localinstall /rubixdbms-policy-base-targeted-6.x.x-x.el6.ixxx.rpm
Only if rubixdbms-policy-base-mls installed
   yum --y localinstall /rubixdbms-policy-devel-mls-6.x.x-x.el6.ixxx.rpm
Only if rubixdbms-policy-base-targeted-installed:
yum --y localinstall /rubixdbms-policy-base-targeted-6.x.x-x.el6.ixxx.rpm
   yum --y localinstall /rubixdbms-policy-devel-mls-6.x.x-x.el6.ixxx.rpm
   yum --y localinstall /rubixdbms-policy-devel-6.x.x-x.el6.ixxx.rpm
   yum --y localinstall /rubixdbms-doc-6.x.x-x.el6.ixxx.rpm
```

These steps will install packages named `rubixdbms`, `rubixdbms-devel`, `rubixdbms-doc`, and `rubixdbms-odbc`. The installation places Trusted RUBIX into the `/var/lib/RUBIXdbms` directory. All user executable programs are accessible from `/usr/bin` as soft links. The ODBC library is accessible from `/usr/lib`. The installation creates the `rubix` user and the `rubixtp` group, which are used to isolate Trusted RUBIX files and processes. All user documentation is accessible in the `/var/lib/RUBIXdbms/doc` directory.

The installation creates two types of SELinux policy modules:

- `rubix-base-mls` and/or `rubix-base-targeted`
- `rubix-dev-mls` and/or `rubix-dev-targeted`

The `rubix-base` module(s) contain all of the policy rules needed for Trusted RUBIX to operate within the RHEL6 operating system and base-rules used by the `rubix-dev` policy module(s). Source code is not included for the `rubix-base` policy module and it may not be altered by the end user.

The `rubix-dev` policy module(s) contain site-specific SELinux policy for RDBMS objects. The install process places the source code for the `rubix-dev` policy into the `/var/lib/RUBIXdbms/etc/selinux` directory. It is intended to be used for on-site custom policy development. The default `rubix-dev` policy module contains policy rules that represent the possible uses of SELinux with Trusted RUBIX and serves to demonstrate its capabilities. It also provides a general purpose SELinux environment for the operation of Trusted RUBIX.

For more information on developing custom security policies for Trusted RUBIX please see the *Trusted RUBIX SELinux Guide*.

© 2014 Infosystems Technology, Inc.
This version of Trusted RUBIX is **not backwards compatible** with databases created with previous versions. Therefore, only databases created with the new version will function with this version of Trusted RUBIX. You may request a database conversion tool that will convert between different versions of Trusted RUBIX. To receive the conversion tool, please email ITI at support@rubix.com with the version numbers of both the new version of Trusted RUBIX and the version of Trusted RUBIX used to create your existing database. You may retrieve the version number of an installed version of Trusted RUBIX by issuing the following command:

```
yum info rubixdbms
```

If the *mcstrans* service is to be used to map user friendly strings to SELinux contexts then the service must be enabled and started. Note that these commands will cause the *mcstrans* service to automatically start with the operating system. Configuring the *mcstrans* service may be accomplished by issuing the following command as the Linux *root* user and *unconfined_r* role:

```
chkconfig mcstrans on
service mcstrans start
```

**TRUSTED RUBIX CLIENT PLATFORM INSTALLATION**

For each platform that will be used as a Trusted RUBIX ODBC client and does not have the *rubixdbms-devel* package installed, install the ODBC client package as follows as the Linux *root* user and *unconfined_r* role (the x's will be replaced by your actual version numbers):

```
cd RPM_PACKAGE_FILE_DIR
yum --y localinstall ./rubixdbms-odbc-6.0.x-x.xxx.xxxx.ixxxx.rpm
```

Take note to install the ODBC client package for your particular client machine. Client packages are located in the *Clients* directory of your installation disk. Note that the *rubixdbms-devel* package includes the ODBC libraries for a server machine, so only install the *rubixdbms-odbc* package on a server machine if the *rubixdbms-devel* package is not installed.

For information on installing the ODBC driver on Microsoft Windows please see the Trusted RUBIX ODBC Guide. Note that the Microsoft Visual C++ 2008 Redistributable Package must be installed on any Windows platform that will host the Trusted RUBIX ODBC Driver. Installation instructions and the package may be found at:


**Configuring Local Communications**

Local connections use UNIX sockets and are labeled automatically by RHEL6 with the context of the client’s process. Therefore, no further configuration is needed. UNIX socket files are created in the `/var/lib/RUBIXdbms/sockets` directory.

To verify a correct installation, it is recommended that Trusted RUBIX first be used in local communications mode before attempting to use Trusted RUBIX in remote communications mode. To use Trusted RUBIX in local communications mode, simply execute your client program (e.g., *rxisql*) on the same machine that contains the Trusted RUBIX server and do not specify a remote host name in the database connection string (e.g., use *MyDB* as opposed to *MyDB@remote.host.com*). Note that other configuration (e.g., role assumption and starting the Trusted RUBIX dispatcher) must be performed prior
to operating Trusted RUBIX as defined later in this document.

**Configuring Remote Communications**

The Trusted RUBIX software uses a single port to communicate with remote clients via INET sockets. The default port number is 4156. If a different port number is desired it may be configured in the `/var/lib/RUBIXdbms/etc/rxconfig` file. If a port other than 4156 is desired, the `dispatcher.listenport` entry in the `rxconfig` file should be added (if it does not exist) or modified as follows:

```
dispenser.listenport = DSPR_PORT_NUMBER
```

If the port number is set to 0 (the number zero) then remote connections to Trusted RUBIX will be disabled. In this configuration only clients on the local host machine will be able to connect to Trusted RUBIX servers. Note that if a port other than the default is used, the port number must be explicitly passed to remote Trusted RUBIX client programs during connection. Local Trusted RUBIX client programs use UNIX sockets and do not require a port number.

The firewall configuration must also allow communication access to the port. This may be accomplished through the GUI Firewall tool from the `System -> Administration -> Firewall` menu. Click on “Other Ports” and then “Add.” Check “User Defined” and add the desired port number using the `tcp` protocol. Click “OK” and then `File -> Apply`.

The Firewall Configuration will look as follows:

![Firewall Configuration](image)

Trusted RUBIX extracts the remote user’s SELinux context directly from the context of the remote socket connection. Therefore, socket connections must be configured with an appropriate SELinux context. This may be accomplished by using `netlabel` or `IPSec`. For more information please see the Trusted RUBIX SELinux Guide and the corresponding RHEL6 documentation.

By default, RHEL6 leaves socket connections from remote hosts unlabeled. Trusted RUBIX assigns the `system_u:object_r:unlabeled_t:s0` context to sessions using unlabeled socket connections. The default `rubix-dev` policy does not permit sessions with the `unlabeled_t` type to connect to a Trusted RUBIX database. Either the `rubix-dev` policy must be altered to allow unlabeled connections (not recommended)
or the connection must be labeled using netlabel or IPSec.

As a simple example, the netlabelctl command (man netlabelctl for more information) may be used to statically assign an SELinux context to unlabeled communications based upon the IP address of the remote host. Note that the netlabel_tools package may need to be first installed on your platform. The following commands (as the root user) will assign different contexts to the unlabeled network traffic depending on if it comes from host 192.168.1.29 or 192.168.1.28.

```
netlabelctl unlbl add default address:192.168.1.29 label:rxdev_u:rubix_remote_client_r:rubix_remote_client_t:s0
netlabelctl unlbl add default address:192.168.1.28 label: rxdev_u:rubix_remote_client_r:rubix_remote_client_t:s1
```

Using the netlabelctl command will apply the mappings only during the current operating system session. A reboot will remove the mappings. To make netlabel mappings permanent place the netlabelctl command arguments into the /etc/netlabel.rules file. For the preceding example, the netlabel.rules file would contain the following two lines (note that the sensitivity of the second entry, `s1`, is only valid for the MLS policy and not the Targeted policy):

```
unlbl add default address:192.168.1.29 label:rxdev_u:rubix_remote_client_r:rubix_remote_client_t:s0
unlbl add default address:192.168.1.28 label: rxdev_u:rubix_remote_client_r:rubix_remote_client_t:s1
```

The netlabel service must be explicitly enabled. The netlabel service may be enabled by using the system-config-services program. The program may be started directly from the command line or through the System->Administration->Services menu. Once the program has started and initialized, the netlabel service may be enabled by scrolling down to the netlabel entry, selecting it with the mouse, and then choosing Enable at the top-left of the program. The netlabel service will now automatically start with every system startup. To start it immediately choose Start at the top of the program.

### Creating a Development User

When performing development of Trusted RUBIX client applications and Trusted RUBIX RDBMS SELinux policy it is beneficial to have a logon user that may access all of the Trusted RUBIX roles. This section describes how to create such a development user. The development user will be able to reach all roles created by the default rubix-dev policy.

For a production environment, it is recommended to partition the roles between users as the security requirements dictate. For information on how to create users for a production environment please see the Trusted RUBIX SELinux Guide.

The steps that follow may be used to create a Linux user that may access all of the Trusted RUBIX roles in both the Targeted and MLS policies. The steps should be executed under the Targeted policy as the Linux root user and unconfined_r role.

1. Create the Linux user rxdev (useradd command or “System->Administration->Users and Groups” menu). It may also be useful to configure the sudo command to allow the rxdev Linux user to execute commands as the root Linux user.

2. Map the rxdev Linux user to the rxdev_u SELinux user for the Targeted and MLS policies. The rxdev_u SELinux user was created when the rubixdbsms-policy-devel package was installed. These commands may also be accomplished using the SELinux Administration GUI tool. It may be found under the “System->Administration->SELinux Management” menu.
3. Logoff the current user and login as the `rxdev` user

4. Change to the desired role (e.g., `newrole -r rubix_dba_r`) and use Trusted RUBIX. You may observe your current security context using the `id –Z` command. You may exit the current role by using the `exit` command. Note that to transition to a Trusted RUBIX role you must first be in the default login role, either the `unconfined_r` role (Targeted policy) or the `user_r` role (MLS policy).

Using the SELinux Administration GUI tool found under the “System->Administration->SELinux Management” menu, the `rxdev_u` SELinux user configuration will look like the following:
The mapping between the `rxdev` Linux user and `rxdev_u` SELinux user will look like the following:

![Image showing SELinux Administration interface]

## Execution and Use

### Roles and Role Transitions

SELinux roles are used to authorize users to perform Trusted RUBIX administrative and SQL operations. Role configuration is performed by the `rubix-dev` SELinux policy module that is installed by the `rubix-policy-devel` package. Default role configurations of the `rubix-dev` policy are described later in this document. This policy may be customized according to local security requirements. For more information on custom SELinux policy configurations please see the [Trusted RUBIX SELinux Guide](#).

Note that despite authorizations assigned to a role via SELinux Type Enforcement, the user still needs to meet the additional requirements of the Discretionary Access Control (DAC), Multilevel Security (MLS), and Attribute Based Access Control (ABAC) security policies.

To use any feature of Trusted RUBIX, including connecting to a database, a user must assume a proper role. Trusted RUBIX roles were created during the installation of the `rubix-dev` policy. Trusted RUBIX roles all have a `rubix` prefix. Additionally, default operating system roles are conditionally given Trusted RUBIX authorizations, based upon SELinux Boolean variables. As discussed latter in this document, Boolean variables may be dynamically set on or off controlling whether the operating system roles have the authorizations.
Roles may be assumed by logging in with the desired role configured as the default or by using the newrole or sudo command. If the newrole command is not available you may need to install the policycoreutils-newrole package. The newrole command may also be used to change the current type or MLS/MCS level.

To transition to a new role there must be SELinux policy rules that allow transition from the source role to the target role. Additionally, the SELinux user must permit the target role. By default, the Trusted RUBIX rubix-dev policy allows role transitions from every operating system login role (e.g., unconfined_r, user_r) to every Trusted RUBIX role (e.g., rubix_client_r, rubix_dbadm_r). Therefore, the distribution of Trusted RUBIX roles may be controlled through the SELinux user configuration. For an example of configuring an SELinux user please see the section titled Creating a Development User.

Typically, a user will login to the system and be assigned a default operating system login role. The default login roles are unconfined_r (Targeted policy) and user_r (MLS policy). The `id –Z` command may be used to determine the current role. The user may then use newrole or sudo to transition to a Trusted RUBIX role. The following commands demonstrate a transition from the unconfined_r role to the rubix_dbadm_r role:

```
[rxdev@RHEL6 ~]$ id -Z
rxdev_u:unconfined_r:unconfined_t:SystemLow-SystemHigh
[rxdev@RHEL6 ~]$ newrole -r rubix_dbadm_r
Password:
[rxdev@RHEL6 ~]$ id -Z
rxdev_u:rubix_dbadm_r:rubix_dbadm_t:SystemLow-SystemHigh
```

OPERATING SYSTEM (OS) ROLES

The Trusted RUBIX rubix-dev policy gives authorizations to operating system login roles. These roles were creating by installing the operating system Targeted and MLS SELinux policies. Therefore, Trusted RUBIX may be used without assuming a Trusted RUBIX role. The authorizations are conditional upon the SELinux Boolean variables, rubix_use_os_client_roles and rubix_use_os_adm_roles. If the variables are set to `on` then the roles will have the Trusted RUBIX authorizations. If the variables are set to `off` then the roles will have no Trusted RUBIX authorizations and the Trusted RUBIX roles must be used. Examples of getting and setting a Boolean value follow (note the root user):

```
[root@RHEL6 ~]# getsebool rubix_use_os_adm_roles
rubix_use_os_adm_roles --on
[root@RHEL6 ~]# setsebool -P rubix_use_os_adm_roles off
```
The following table gives an overview of the Type Enforcement authorizations given to login operating system roles by the default *rubix-dev* policy.

<table>
<thead>
<tr>
<th>Operating System Role</th>
<th>Operations Permitted by Type Enforcement (MLS &amp; DAC still enforced)</th>
<th>Boolean Variable Used to Control Authorizations</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>unconfined_r</strong></td>
<td>→ Client connect (<em>rxisql</em> &amp; <em>ODBC</em>) &lt;br&gt; → All SQL DML &amp; DDL operations for <em>default</em> object set &lt;br&gt; → Import &amp; export (<em>rximport/rxexport</em>)</td>
<td>→ <em>rubix_use_os_client_roles</em></td>
</tr>
<tr>
<td><strong>user_r</strong></td>
<td>→ Client connect (<em>rxisql</em> &amp; <em>ODBC</em>) &lt;br&gt; → All SQL DML &amp; DDL operations for <em>default</em> object set &lt;br&gt; → Import &amp; export (<em>rximport/rxexport</em>)</td>
<td>→ <em>rubix_use_os_client_roles</em></td>
</tr>
<tr>
<td><strong>sysadm_r</strong></td>
<td>→ Client connect (<em>rxisql</em> &amp; <em>ODBC</em>) &lt;br&gt; → All SQL DML &amp; DDL operations for <em>default</em> and <em>objset1</em> object sets &lt;br&gt; → <em>CREATE CATALOG</em> for <em>default</em> object set &lt;br&gt; → Import &amp; export (<em>rximport/rxexport</em>) &lt;br&gt; → Create database (<em>rxisql</em>) &lt;br&gt; → Drop database (<em>rxdb</em>) &lt;br&gt; → Backup database (<em>rxdump</em>) &lt;br&gt; → Restore databases (<em>rxxrestore</em>) &lt;br&gt; → Supersede the SQL DAC policy &lt;br&gt; → Operate dispatcher (<em>rxjsvrman</em>) &lt;br&gt; → Terminate servers (<em>rxjsvrman</em>) &lt;br&gt; → <em>SPM policy management</em> (<em>rxpolman</em>) &lt;br&gt; → Session context assignment &lt;br&gt; → Object context assignment &lt;br&gt; → Configure auditing (<em>rxauditset</em>) &lt;br&gt; → Read audit trail (<em>rxauditrpt</em>)</td>
<td>→ <em>rubix_use_os_adm_roles</em></td>
</tr>
<tr>
<td><strong>sysadm_r</strong> (MLS policy only)</td>
<td>→ Client connect (<em>rxisql</em> &amp; <em>ODBC</em>) &lt;br&gt; → All SQL DML &amp; DDL operations for <em>default</em> object set &lt;br&gt; → <em>CREATE CATALOG</em> for <em>default</em> object set &lt;br&gt; → Import &amp; export (<em>rximport/rxexport</em>) &lt;br&gt; → Create database (<em>rxisql</em>) &lt;br&gt; → Drop database (<em>rxdb</em>) &lt;br&gt; → Backup database (<em>rxdump</em>) &lt;br&gt; → Restore databases (<em>rxxrestore</em>) &lt;br&gt; → Supersede the SQL DAC policy &lt;br&gt; → Operate dispatcher (<em>rxjsvrman</em>) &lt;br&gt; → Terminate servers (<em>rxjsvrman</em>)</td>
<td>→ <em>rubix_use_os_adm_roles</em></td>
</tr>
<tr>
<td><strong>staff_r</strong></td>
<td>→ Client connect (<em>rxisql</em> &amp; <em>ODBC</em>) &lt;br&gt; → All SQL DML &amp; DDL operations for <em>default</em> object set &lt;br&gt; → Import &amp; export (<em>rximport/rxexport</em>) &lt;br&gt; → Backup database (<em>rxdump</em>) &lt;br&gt; → Operate dispatcher (<em>rxjsvrman</em>) &lt;br&gt; → Terminate servers (<em>rxjsvrman</em>)</td>
<td>→ <em>rubix_use_os_adm_roles</em></td>
</tr>
<tr>
<td><strong>auditadm_r</strong> (MLS policy only)</td>
<td>→ Client connect (<em>rxisql</em> &amp; <em>ODBC</em>) &lt;br&gt; → All SQL DML &amp; DDL operations for <em>default</em> object set &lt;br&gt; → Import &amp; export (<em>rximport/rxexport</em>) &lt;br&gt; → Configure auditing (<em>rxauditset</em>) &lt;br&gt; → Read audit trail (<em>rxauditrpt</em>)</td>
<td>→ <em>rubix_use_os_adm_roles</em></td>
</tr>
</tbody>
</table>
Operating System Role | Operations Permitted by Type Enforcement (MLS & DAC still enforced) | Boolean Variable Used to Control Authorizations
---|---|---
secadm_r (MLS policy only) | ➔ Client connect (rxisql & ODBC) | ➔ rubix_use_os_adm_roles
 | ➔ All SQL DML & DDL operations for default object set | |
 | ➔ Import & export (rximport/rxexport) | |
 | ➔ SPM policy management (rxpolman) | |
 | ➔ Session context assignment | |
 | ➔ Object context assignment | |

**TRUSTED RUBIX ROLES**

Trusted RUBIX roles are created by the *rubix-dev* policy during the installation of *rubix-policy-devel* package. The roles have a `rubix` prefix and may be used to operate the Trusted RUBIX RDBMS. In general, roles are created to serve as a client, database administrator, security administrator, audit administrator, and operator. The following table gives an overview of the Trusted RUBIX roles created by the default *rubix-dev* policy.

<table>
<thead>
<tr>
<th>Trusted RUBIX Role</th>
<th>Operations Permitted by Type Enforcement (MLS &amp; DAC still enforced)</th>
<th>Reachable From Role</th>
</tr>
</thead>
</table>
rubix_dbadm_r | ➔ Client connect (rxisql & ODBC) | ➔ unconfined_r
 | ➔ All SQL DML & DDL operations for default and objset1 object sets | ➔ user_r
 | ➔ CREATE CATALOG for default object set | ➔ staff_r
 | ➔ Import & export (rximport/rxexport) | ➔ sysadm_r
 | ➔ Create database (rxisql) | ➔ secadm_r (MLS Policy)
 | ➔ Drop database (rxdb) | ➔ auditadm_r (MLS Policy)
 | ➔ Backup database (rxdump) | |
 | ➔ Restore databases (rxrestore) | |
 | ➔ Supersede the SQL DAC policy | |
 | ➔ Operate dispatcher (rxsvrman) | |
 | ➔ Terminate servers (rxsvrman) | |
rubix_secadm_r | ➔ Client connect (rxisql & ODBC) | ➔ unconfined_r
 | ➔ All SQL DML & DDL operations for default object set | ➔ user_r
 | ➔ SELECT for objset1 object set | ➔ staff_r
 | ➔ Import & export (rximport/rxexport) | ➔ sysadm_r
 | ➔ SPM policy management (rxpolman) | ➔ secadm_r (MLS Policy)
 | ➔ Session context assignment | ➔ auditadm_r (MLS Policy)
 | ➔ Object context assignment | |
rubix_auditadm_r | ➔ Client connect (rxisql & ODBC) | ➔ unconfined_r
 | ➔ All SQL DML & DDL operations for default object set | ➔ user_r
 | ➔ SELECT for objset1 object set | ➔ staff_r
 | ➔ Import & export (rximport/rxexport) | ➔ sysadm_r
 | ➔ Configure auditing (rxauditset) | ➔ secadm_r (MLS Policy)
 | ➔ Read audit trail (rxauditrpt) | ➔ auditadm_r (MLS Policy)
 | ➔ Operate dispatcher (rxsvrman) | |
rubix_op_r | ➔ Client connect (rxisql & ODBC) | ➔ unconfined_r
 | ➔ All SQL DML & DDL operations for default object set | ➔ user_r
 | ➔ SELECT for objset1 object set | ➔ staff_r
 | ➔ Import & export (rximport/rxexport) | ➔ sysadm_r
 | ➔ Backup database (rxdump) | ➔ secadm_r (MLS Policy)
 | ➔ Operate dispatcher (rxsvrman) | ➔ auditadm_r (MLS Policy)
 | ➔ Terminate servers (rxsvrman) | |
### Starting the Trusted RUBIX Dispatcher

Prior to initiating database sessions the `rubixdbms` service (`rxdspr` process) must be started. By default, the `rubixdbms` service will start during the operating system boot. Explicitly controlling the service is done using the `rsxvrman` Trusted RUBIX command or the `service` operating system command. The `rsxvrman` command must be performed by a user currently operating in the `rubix_op_r`, `rubix_dbadm_r`, `sysadm_r`, or `staff_r` role. The `service` command must be performed by the `root` Linux user in the `unconfined_r` role. Note that the `rsxvrman` command may also be used to control currently instantiated RDBMS servers (`rxserver` process) and the maintenance server (`rxmsvr` process). The dispatcher may be started as follows:

```
rsxvrman -s
```

or

```
service rubixdbms start
```

It may be restarted as follows:

```
rsxvrman -r
```

or

```
service rubixdbms restart
```
The dispatcher and all instantiated, idle servers may be terminated as follows:

```
rxsvrman -t
```
or
```
service rubixdbms stop
```

The status of the service may be retrieved as follows:

```
service rubixdbms status
```

## Creating a Database

Prior to performing SQL operations a database must be created. This may be done using the `rxisql` command. The `rxisql` client is a text based command tool used to submit SQL operations to the server software. Note that the command must be performed by a user currently operating in the `rubix_dbadm_r` or `sysadm_r` role. The database may be created as follows (the examples assume a local connection):

```
rxisql -d master
rxsql> create database MyDB;
rxsql> q
```

Note that the `master` database is used to create other databases. The `master` database itself need not be created. Additionally, if you wish to specify the storage directories for the database logs and table data please consult the **Trusted RUBIX Trusted Facility Manual**.

The database may be dropped using the `rxdb` command by those in the `rubix_dbadm_r` or `sysadm_r` role as follows:

```
rxdb -d MyDB
```

## Performing Client SQL Operations

The `rxisql` client is a text based command tool used to submit SQL operations to the server software. Note that the command must be performed by a user currently operating in the `rubix_client_r`, `unconfined_r`, `user_r` role or any of the administrative roles. Note that the database connection string (argument to the `-d` option) contains the database name only (e.g., `MyDB`) for local connections and a remote host and optional port number for remote connections (e.g., `MyDB@my.host.com:4156`) The `rxisql` command may be used to perform SQL operations as follows (the examples assume a local connection):

```
rxisql -d MyDB
rxsql> create table MyTable (col1 int);
rxsql> insert into MyTable values 555;
rxsql> commit;
rxsql> q
```

## Configuring and Creating ODBC Applications

ODBC applications are custom programs that may access standards compliant RDBMS’s. ODBC is Microsoft’s adaptation of the X Open CLI standard. The following website gives details of its use, including the API specification:
Please refer to the Trusted RUBIX ODBC Guide for details about building applications with ODBC and configuring the Trusted RUBIX ODBC client to interoperate with pre-existing ODBC compliant applications.

Uninstalling Trusted RUBIX

To completely uninstall Trusted RUBIX remove the installed packages and then recursively remove the /var/lib/RUBIXdbms directory. If you wish to retain the configuration and database files do not remove the /var/lib/RUBIXdbms directory. To erase the Trusted RUBIX packages become the root user and assume the unconfined_r role and issue the following commands in the given order:

```
yum remove rubixdbms-doc
yum remove rubixdbms-devel
yum remove rubixdbms
yum remove rubixdbms-policy-devel
yum remove rubixdbms-policy-base
yum remove rubixdbms-odbc
```

Overview of User Documentation

All user documentation is in PDF format and is found in the /var/lib/RUBIXdbms/doc directory. The following is a brief overview of the documentation.

- **RX_SELinux_Guide_6_0.pdf**
  The SELinux Guide describes the SELinux security mechanism as it relates to Trusted RUBIX and how to create custom SELinux DBMS policies.

- **RX_Trusted_Facility_Manual_6_0.pdf**
  The Trusted Facility Manual describes administrative aspects of the DAC and MAC security policies.

- **RX_Security_Features_Users_Guide_6_0.pdf**
  The Security Features User’s Guide describes aspects of the DAC and MAC security policies as they relate to normal database user operations.

- **RX_Commands_6_0.pdf**
  The Administrative Commands Reference Guide contains instructions on the proper use of the administrative commands and their security behaviors.

- **RX_SQL_Guide_6_0.pdf**
  The SQL Reference Guide contains the details on the use of the SQL language specification.

- **RX_ODBC_Guide_6_0.pdf**
  The ODBC Guide contains the details of the ODBC programming specification, application build instructions, and configuration information.

- **RX_Information_Schema_6_0.pdf**
  The Information Schema Guide describes all schema views and their security behavior.

- **RX_Security_Policy_Manager_Reference_Guide_6_0.pdf**
that is based on X Open’s XACML standard.

→ **RX_Application_User_Guide_6.0.pdf**
  The Application User Guide describes the use and deployment of the Trusted RUBIX Application User mechanism. This mechanism extends the Attribute Based Access Control (ABAC) of the Security Policy Manager to external (e.g., Internet) users of Trusted RUBIX applications.

→ **RX_Security_Policy_Manager_Tutorial_6.0.pdf**
  The Security Policy Manager Tutorial gives step-by-step examples of building and using the SPML language to create a variety of tailored security policies.

→ **RX_SQL_Tutorial_6.0.pdf**
  The SQL-ODBC Tutorial gives step-by-step examples to using Structure Query Language (SQL) and building ODBC applications.

**Support and Further Information**

The most recent versions of the documentation are posted at [http://www.rubix.com/](http://www.rubix.com/) for download. Support and further information may be acquired by email at support@rubix.com.