Upgrading Rhythmyx Version 4.5 to Version 5.7

Introduction

Upgrading from Rhythmyx Version 4.5 to Version 5.7 involves incorporating a variety of features that were introduced in Versions 5.0, 5.5 and 5.6. Upgrading to Version 5.0 features in particular is a substantial project, and requires extensive planning and preparation. This document will help guide you through the upgrade project.

The first section of this document outlines the impact of the upgrade on system infrastructure.

The second section of the document delineates the planning and practices recommended to complete the upgrade successfully.

The third section of this document includes a checklist of modifications to the Rhythmyx server in Version 5.0 and 5.5 that may prevent the server from functioning correctly following upgrade, or may require reimplementation to restore system stability and existing functionality. Some of these modifications may not affect your system, but a few modifications affect all systems. The subsection following the checklist provides details about the modifications required to restore stability and functionality.

The final section of this document is a checklist of new features added in Rhythmyx Version 5.0, Version 5.6 and Version 5.7, followed by details about implementation of these new features.

Rhythmyx Version 5.7 Upgrade and Your System Infrastructure

Hardware Requirements for the Version 5.7 Upgrade

The only new feature in Rhythmyx 5.7 that has a potential impact on hardware requirements is the full-text search engine. The full-text search engine adds roughly 100 MB to the disk space requirements ruing both installation and operation. In operation, the full-text search engine may also increase RAM usage. Since full-text searches on very large repositories may have a significant adverse impact on performance, Percussion software offers three options when upgrading:

- The default option is to install the full-text search engine on the same machine as the Rhythmyx server. This option requires an additional disk storage space during operation. Operation of the search engine for small repositories may require as little as 100 MB above the disk storage requirements of the Rhythmyx server during operation, but larger repositories need additional space; Percussion Software recommends disk space for the search engine equal to the disk space available for the repository itself for larger systems. In addition to the disk requirements, the search engine requires three additional port listeners beyond the port used by the Rhythmyx server. (The default ports are 9993, 9994, and 9995).
- Customers with very large repositories, or whose Rhythmyx server currently operates near its functional limits may want to consider a remote installation of the full-text search engine on a separate machine. This option requires some additional manual setup, but it minimizes the effect of the full-text search engine

on the Rhythmyx server and dedicates processing power to the search engine itself. If you choose to co-locate the full-text search engine with the Rhythmyx server, you can still choose to move the search engine to another machine later, although this process does require some manual setup.

- For customers whose Rhythmyx Version 5.0 server is already operating at or over capacity, the option of not installing the full-text search engine is also available. In this case, full-text search capabilities will not be available (nor will the new text extraction functionality); the lightweight, database search engine available with previous version will still function, however. The overall processing demand of the Rhythmyx server using this option is equal to that of a Version 5.0 server.

WebDAV and Rhythmyx Hardware Requirements for the Rhythmyx Repository

Customers considering the use of WebDAV functionality should take into account the potential impact of greatly increased volume of repository database. WebDAV makes it possible to drag and drop hundreds of binary attachments into the repository quickly and easily. The addition of a significant new volume of binary data may demand greatly increased storage space and may adversely affect database performance. Modern database management systems generally store binary data much more efficiently than earlier version of these systems; nevertheless, customers should be aware the volume of binary data that can be added to the database using WebDAV may noticeably degrade the performance of the RDBMS.

Platform Changes Between Rhythmyx Version 4.5 and Version 5.7

Percussion Software has made some minor changes in supported platforms and operating systems to keep up with newer versions of the platforms and operating systems. Please remember that these changes are to the platforms and operating systems that our Software Quality Assurance team certifies during the testing process.

Operating Systems

Rhythmyx is certified on three operating system platforms, Microsoft Windows, Sun Solaris, and Linux.

Microsoft Windows Operating System

Support for Windows NT has been deprecated. The Rhythmyx server is certified on Windows 2000 and Windows 2003 server lines.

Sun Solaris Operating System

The Rhythmyx server is certified on Sun Solaris SPARC Versions 7, 8, and 9.

Linux Operating System

In Version 5.7, the Rhythmyx server is certified on Red Hat AS 3.0. Support is available for other versions of the Linux operating system, but Red Hat AS 3.0 is the only Linux operating system certified.

Rhythmyx Client Certification

Rhythmyx clients include the Workbench, Server Administrator and Multi-server Manager. Support for these clients on Windows NT is deprecated. Rhythmyx clients are supported on Windows 2000, Windows XP, and Windows Server 2003 operating systems.

Relational Database Management Systems (RDBMS)

Rhythmyx continues to support the following relational database management systems:

- Oracle
  - 8i
  - 9i
  - (support for Oracle 10g is pending the last quarter of 2004)
  - (Support for Oracle 7 is deprecated)
- Microsoft SQL Server
  - SQL Server 2000
  - (Support for SQL Server 7 is deprecated)
- Sybase
  - Version 12.5
- IBM DB2 UDB
  - Version 7.2 and later

Supported Browsers

Microsoft Internet Explorer 5.5 and later are certified. Internet Explorer Version 5.0 is no longer supported.

Netscape Communicator 7.0 is certified on Microsoft Windows. Netscape 4.x is no longer supported. Mozilla 1.5 and higher are supported on Microsoft Windows.

Both Netscape and Mozilla require the open-source Esker plugin which enables ActiveX support, to support the rich-text editing control. The Esker plugin is available from Esker plugin support site at Yahoo Groups (http://groups.yahoo.com/group/esker-activex-plugin/).

Apple Macintosh Support

Rhythmyx Version 5.7 supports the Safari 1.21 browser on the Apple OS/X 10.3 and high operating systems. The Ektron eWebEditPro rich text editor control is not supported on this platform, however. If rich text editing is as mission-critical feature, please contact your Percussion Software sales representative to discuss your options.

Portal Connector

The BEA WebLogic connector that ships with Rhythmyx 5.7 is certified with WebLogic Portal 8.1. Support for WebLogic 7.x is deprecated.

Java Runtime Engine Browser Plugin

Rhythmyx Version 5.7 requires a Java runtime engine browser plugin to operate the Content Explorer applet. The minimum recommended version of this plugin is Version 1.4.2. Version 1.5 is the minimum recommended version to provide full support for accessibility in Rhythmyx.

Accessibility

New features for product accessibility are certified on Internet Explorer 6. In addition, certain accessibility features in the Content Explorer require bug fixes that Sun has included in the JSRE Java plugin Version 1.5 and in a new version of the Java Accessibility Bridge components. Version 1.5 of the Java plugin is the minimum requirement for full accessibility compliance.

Upgrade Installation Code

Upgrade to Rhythmyx 5.7 requires a new installation code. The Version 5.7 installer will not recognize a Version 4.5 installation code as valid, and the installation will fail.

Recommended Planning and Practices

Upgrading Enterprise Software can be a complex project. Architecture changes in the new version often require customizations made on the previous version to be tweaked or even re-architected in the new version. This guide will assist you in identifying components in your existing implementation affected by upgrading to Rhythmyx 5.7, and help you develop an upgrade strategy to deliver the new functionality with minimal impact to end users.

Rhythmyx 5.0 introduced new best practices around the development model in addition to new features. This guide breaks the upgrade process into 4 distinct phases:

- Planning
- Development
- Integration
- Production

The upgrade process conforms to these new practices, and will accomplish 2 goals:

- Upgrade the Rhythmyx environment from Rhythmyx 4.x to Rhythmyx 5
- Create a multi-tiered Rhythmyx environment leveraging Multi-Server Manager to allow distinct activities to occur on each tier.

Rhythmyx 5.7 Upgrade Overview

I. Prerequisites
   1. Rhythmyx Version: 4.5 build 20020621 or greater

2. Current Development Work
    a. Code freeze on production server
    b. 2-4 weeks of stability
3. Backup and Recovery Plan
4. Rhythmyx 5.7 Training
    a. Rhythmyx 5.7 Developer 1
    b. What’s new in Rhythmyx 5.7
5. Rhythmyx License
    a. Rhythmyx Multi Server Manager
    b. Integration Server
    c. User Acceptance Testing
6. Hardware Requirements
7. Database Requirements

II. Planning Phase

1. What’s New in Rhythmyx 5.0, 5.5, 5.6, and 5.7?
2. Identifying Implementation elements and Assessing Upgrade Impact
    a. Identifying implementation elements that require no modification
    b. Identifying implementation elements that require modification for Rhythmyx 5.0 and 5.7 requirements
3. Examining new features
4. Multi-Tiered Rhythmyx Environment
    a. Developing a Multi-Server Manager strategy
5. Assessing End User Impact
6. CMS Availability

III. Upgrade Development Phase

1. Create a production replica
    a. Leverage existing hardware
2. Run the upgrade installation
3. Perform any system modifications determined necessary
    a. Document the system modification process
4. Migrate 4.5 implementation elements to Rhythmyx 5.7 specifications
    a. Walk through the Upgrade Checklist
    b. Create a detailed log of all implementation upgrade work
5. Create MSM archives
6. Maintaining the current Production Environment during the Upgrade process

IV. Upgrade Integration Phase

1. Create a Production Replica
2. Run the Upgrade Installation
3. Backup the Upgrade Installation Server
4. Apply system modifications
5. Install MSM archives created during Development Upgrade Phase
6. QA Testing
7. UAT Testing
8. Develop User Documentation
9. Develop User Training

10. Perform Pre-Production User Training

V. Upgrade Production Phase

1. Backup the Production Server
2. Schedule Production Server Downtime
3. Run the Upgrade Installation
4. Apply system modifications
5. Install MSM archives created during the Development Phase and tested during the Integration Phase.
6. QA Testing
7. Distribute User Documentation
8. Perform User Training

VI. Post-Upgrade

1. Deprecate old Development and Integration environments
2. Use the new Multi-Tiered Development practices
3. Plan and scope new Rhythmyx 5 development work

Prerequisites

Rhythmyx Version

The Rhythmyx 5.7 upgrade requires at a minimum build 20020621 or greater. If the build of Rhythmyx Version 4.5 you have installed is more recent that this build, you can upgrade directly to Rhythmyx Version 5.7. If the build of Rhythmyx Version 4.5 you have installed is older than 20020621, you must upgrade to a build of Rhythmyx Version 4.5 more recent than 20020621 before upgrading to Rhythmyx Version 5.7.

Rhythmyx Version 4.0 cannot be upgraded to Rhythmyx Version 5.7. You must perform an interim upgrade to Rhythmyx 5.01 before upgrading to Rhythmyx Version 5.7.

Current Development Work

Finish all current development projects before upgrading. Percussion Software recommends a 4-week period of stability after a major development project before performing a Rhythmyx upgrade. Planning work may be done during this time period.

Backup and Recovery Plan

Percussion Software strongly recommends regular backup of all Rhythmyx production environments, and ideally of development and integration servers as well. In preparation for upgrading, completely back up your production environment to ensure that you have a stable environment to fall back on in case you encounter problems during the upgrade process.

Rhythmyx 5.7 Training

Due to the number of new features introduced in Rhythmyx 5.0, Percussion Software highly recommends developer training on Rhythmyx 5.7 before attempting an upgrade.

- Developer 1 Training in Rhythmyx 5.7 – This course covers the same scope as Rhythmyx Developer 1 training in previous versions, updated for Rhythmyx 5.7. This class integrates Multi-Server Manager into each module. Developers new to

Rhythmyx, or developers interested in a refresher course benefit the most from this course.

- What's New in Rhythmyx 5.7 – This short course covers all the new features in Rhythmyx 5.7 except Multi-Server Manager. Experienced developers

Rhythmyx 5.7 Licensing

The recommended practice for upgrading to Rhythmyx 5.7 requires a multi-tiered Rhythmyx environment. This will require 2 server licenses in addition to the Production Server. All servers required for the upgrade can be installed with either of the following licenses. There are subtle differences in regards to the number of developers that can work on the server simultaneously, how many users can connect simultaneously during user acceptance testing. Contact your account manager regarding any licensing questions.

- Developer License – allows a single developer or tester to connect to a Rhythmyx Server.
- Test License – allows multiple developers and testers to connect to a Rhythmyx Server simultaneously.

Additionally, Multi-Server Manager requires a special license in order for this client to connect to the Rhythmyx Server. Talk to your Account Manager to obtain this license.

Hardware Requirements

The upgrade process involves creating 2 Rhythmyx installations for Development and Integration. These installations do not require new hardware, and can be installed on existing servers. The setup steps in this document include special instructions for leveraging existing hardware.

Database Requirements

The upgrade process involves creating 2 Rhythmyx installations for Development and Integration. Each installation requires its own separate repository, but multiple repositories can exist within a database server. A repository requires a unique schema within the database.

Planning Phase

What's new in Rhythmyx 5.7?

Assessing the upgrade impact requires familiarity with new features in Rhythmyx. Before upgrading, read the What's New documents for Rhythmyx 5.0, 5.5, and 5.7 from the Percussion Software Web Site. Some new features receive detailed coverage in the following guides (available on the Rhythmyx CD as well as our website [http://www.percussion.com/support/Rhythmyx/RxDocumentation.htm]):

- Implementing Content Editors.pdf
- Implementing the Rhythmyx Business User's Interface 5_7.pdf
- Implementing the Relationship Engine 5_7.pdf
- Internationalizing and Localizing Rhythmyx 5_7.pdf

- Maintaining Workflows in Rhythmyx 5_7.pdf
- Rhythmyx Multi-Server Manager 5_7.pdf
- Web Services API Reference.pdf
- Implementing WebDAV in Rhythmyx.pdf

Percussion Software also recommends installing Rhythmyx 5.7 on a development server as a new installation. This optional exercise provides developers an environment to experiment with new Rhythmyx 5.0, 5.5, 5.6, and 5.7 features to improve understanding before upgrading. Do not upgrade a Rhythmyx 4.5 development server. You can use a development license for this server.

Identifying Implementation Elements and Assessing Upgrade Impact

Successful upgrades depend on proper identification of all implementation specific Rhythmyx elements impacted by the upgrade to Rhythmyx 5.7. Upgrade impact assessment involves evaluating the current implementation, identifying customized elements, assessing the impact Rhythmyx 5.7 on each, and developing migration and testing plans for each.

Identifying customized Rhythmyx elements requires knowledge of the implementation. Review the Statement of Work if provided by Percussion Software Professional Services or a third party for any implementation work. This document should detail many of the Rhythmyx elements specific to your implementation. Read any other implementation specific documentation, including developer logs and end-user documentation.

Assessing the impact requires a basic understanding of the new functionality in Rhythmyx 5.0, 5.5, 5.6, and 5.7. Read this documentation completely, paying special attention to the final section, which details the new features requiring additional implementation.

Your implementation may include undocumented changes, or the documentation could be out of date. Regardless, this documentation provides a good starting point for assessing the affect of the upgrade on your system. Also review the checklists later in this document to assist in upgrade impact assessment. For each element listed in the checklist, review the implementation specific documentation and examine the implementation of each occurrence of that element.

Create an implementation specific checklist with details including:

- Element Type (e.g. Content Type)
- Element Name (e.g. Article)
- Element ID (e.g. 301)
- Migration Plan – the steps needed to make this element Rhythmyx 5.5 compliant
- Test Plan – not all customizations will require migration plans, but should be tested anyway.

Note on customizations to system elements: Upgrade installations overwrite all system elements, effectively removing any system customizations. For this reason, system elements are rarely customized. If a system application has been customized, review the customization to understand why it was implemented.

- If the customization was a patch or a bug fix, only re-apply the patch if the bug still exists. Contact Percussion Technical Support to determine if a bug has been addressed in your build. Also, include a test case for the QA cycle after upgrading.
- If the customization was implemented to provide functionality that addressed a limitation in the product, reapply the customization only if Rhythmyx 5.7 does not address the limitation. Contact Percussion Technical Support to determine whether Rhythmyx 5.7 functionality corrects the previous limitation.
- If the customization was implemented to provide a cosmetic change, first determine whether a Rhythmyx 5.7 feature alleviates the need for this customization. If so, then use the Rhythmyx 5.7 functionality. If not, work with Percussion Technical Support to determine whether this type of customization is possible, and if so, the best way to implement it in Rhythmyx 5.7.

Additionally, the architecture of Rhythmyx 5.7 may require a customization to be applied in a different manner. Contact Percussion Technical Support to determine the best way to address this issue.

New Rhythmyx 5.7 Features

The upgrade project should focus on the migration of Rhythmyx 4.5 implementation elements to a Rhythmyx 5.7 environment. Some implementation elements will be completely re-designed using Rhythmyx 5.7 functionality; others will remain much the same. To simplify the upgrade process, Percussion Software recommends that any new Rhythmyx 5.7 development work be scheduled after completing the Rhythmyx 5.7 upgrade.

Multi-Tiered Rhythmyx Environment

Multi-Server Manager is crucial to a successful Rhythmyx 5.7 upgrade and facilitates implementation of the preferred three-tiered CMS environment. This environment structure segregates the tasks of CMS development, testing, and content maintenance into discrete blocks. For details about this environment, see “Multi-Tiered Environment” in Modeling a Content Management System 5.7.pdf. Multi-Server Manager acts as a bridge between the three tiers. Upgrading through a process that creates these three tiers minimizes production server downtime. Using Multi-Server Manager requires detailed knowledge of the Rhythmyx implementation. Read Rhythmyx Multi Server Manager 5.7.pdf and follow the recommended practices in that guide to develop a plan for creating archives to move upgraded Rhythmyx elements between tiers.

Assessing End User Impact

Rhythmyx Version 5.0, 5.5, and 5.6 contained many new features, most significantly, the Content Explorer interface. End users will need to learn many new features, and different ways of completing their normal tasks in Content Explorer. The Rhythmyx 5.7 upgrade impacts the Business Users in the following ways:

- Client Environment
    - JVM
      The client environment now requires a JVM for accessing the Content Explorer Java Applet. Standard user configurations should be tested to ensure compatibility. Rhythmyx requires a minimum JVM of 1.4.2_04. Rhythmyx 5.7 allows administrators to specify a minimum or exact JVM for each OS/Browser combination.

      Static JVM requirements will force users to use a specific JVM while accessing Rhythmyx. Dynamic JVM requirements will enforce a minimum JVM version.
    - Bookmarks
      All Rhythmyx users should be advised to modify any bookmarks to point to http://server:port/Rhythmyx. As an additional precaution, we recommend turning off the application sys_ca to prevent users from accessing the old interface (which will not work seamlessly with Rhythmyx 5.7, but may be useful for testing).
- Documentation
  Business user documentation is built into the Content Explorer interface. Each element of the implementation that works differently under Rhythmyx 5.7 should include pointers to the appropriate Business user documentation topic.
- Training
  End users may require training on the new UI enhancements, as well as any new features implemented (e.g. Workflow enhancements, Communities, Folder Structure, etc…). This training should be prepared before the final upgrade against the production server.
- User Acceptance Testing
  Before exposing the upgraded system to the entire user base, Percussion Software strongly recommends performing User Acceptance Testing. This testing will help find bugs in the Rhythmyx applications, refine usability, and pinpoint areas that will require additional end user documentation and training. The rollout will begin with a few users who already know the new interface and can help others get up to speed quickly. Plan for at least a full week of end user testing as part of the upgrade process. Note: User Acceptance Testing requires a Test Development server license. Consult your account manager with any license related questions, or refer to the *Modeling a Content Management System 5.7.pdf* document for details on Rhythmyx licensing.

Planning for System Availability during the Upgrade

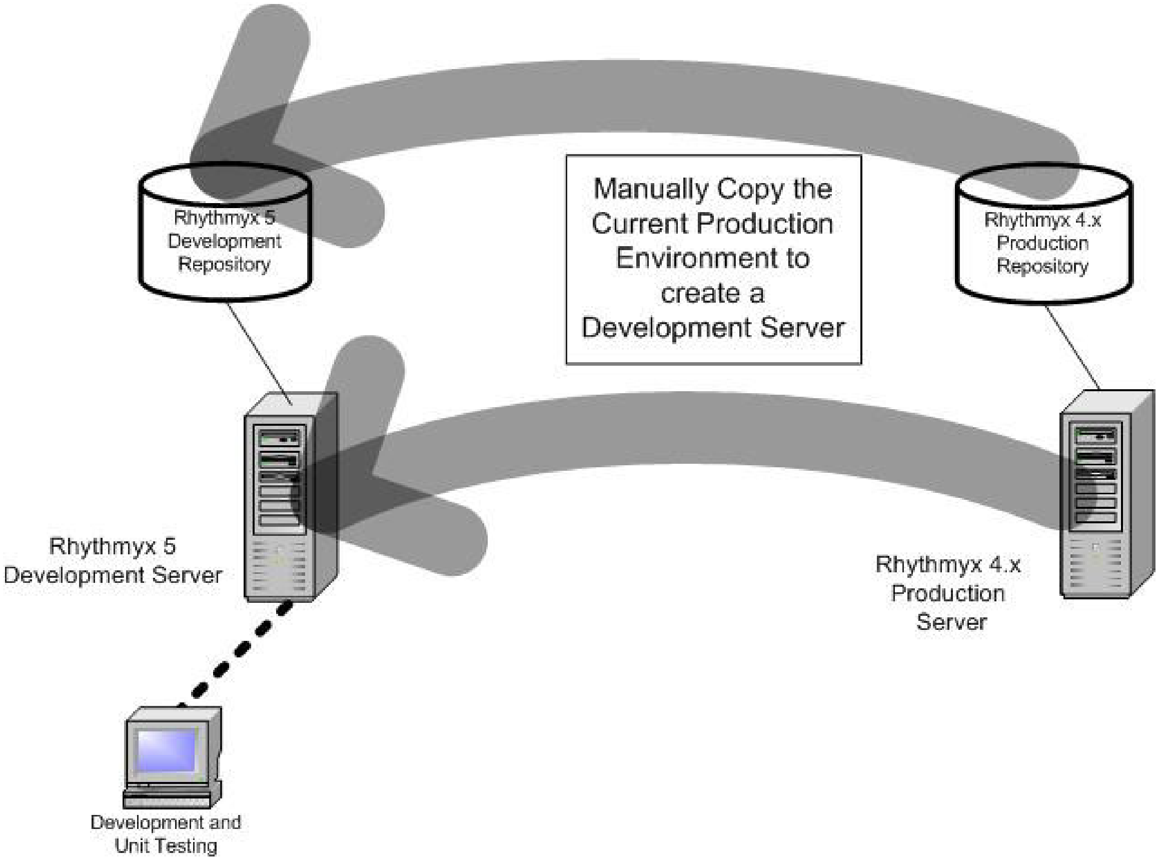
The Rhythmyx 5.7 upgrade project can take several weeks, potentially even a few months, depending on the size and complexity of your implementation, the resources available for the upgrade project, the skill level of your developers, and other factors. The multi-tiered Rhythmyx environment recommended in this document will help you maintain the availability of your Rhythmyx production environment during the upgrade project. Each server, which represents a logical tier in the system, is upgraded and tested before proceeding to the next tier.

To maintain the availability of your Rhythmyx CMS during the upgrade process, you should maintain two concurrent Rhythmyx environments: the development, integration, and production servers of the current Rhythmyx 4.5 environment and the development and integration servers of the Rhythmyx 5.7 upgrade environment. You do not need to run the development and integration servers for both environments simultaneously, but you should have all of them available to start as needed. After you complete the upgrade project, you can deprecate the Rhythmyx 4.5 development and integration environments.

Upgrade Development Phase

The Development Upgrade Phase involves creating a development server for Rhythmyx 5.7 that matches all functional aspects of the Development Tier discussed in Modeling a Content Management System 5.7.pdf (review this document before continuing).

Figure 1 displays the Rhythmyx upgrade environment and activities during the Upgrade Development Phase


Manually Copy the Current Production Environment to create a Development Server
Rhythmyx 5 Development Repository
Rhythmyx 4.x Production Repository
Rhythmyx 5 Development Server
Rhythmyx 4.x Production Server
Development and Unit Testing

Creating a Production Replica

The upgrade development server should start out as a clone of the production Rhythmyx 4.5 server. (Note: To maintain consistency, remember to apply any production changes during the upgrade project to both the current production server AND the upgrade development server).

If using existing hardware with a Rhythmyx server supporting the current Rhythmyx 4.5 environment already installed, follow these pre-setup steps:

1. Backup the Rhythmyx Server
2. Shut-down the Rhythmyx Server
3. Rename the Rhythmyx Directory (e.g. Rhythmyx-OLD)

Follow these steps to create a new Rhythmyx server based on the current Production environment:

1. Create a copy of the Production Repository
    a. Use the same database password to simplify this process; Rhythmyx uses encrypted passwords in all configuration files by default. If you require a

different database password, contact Percussion Software Technical Support for assistance encrypting the password.

2. Create a copy of the Production file tree
    a. Percussion Software recommends moving this server to a different physical location than the current production server.
    b. If using Solaris
        i. Move the Rhythmyx Server to a new user's home directory, or another Solaris server.
        ii. Make sure that the owner of the UNIX directory also owns the Rhythmyx directory
            1. # chown -R [UNIX user] [Rhythmyx Home]
3. Make sure the following configuration files point to the new database. If they do not, setDatabase may encounter errors, the server will not start up properly, or an upgrade installation could accidentally run against the production repository. **DO NOT PROCEED TO THE NEXT STEP UNTIL THESE FILES ALL POINT TO THE NEW REPOSITORY.**
    a. rxconfig/Workflow/rxworkflow.properties
        i. DB_SERVER=[Database Server Name or IP Address]
        ii. DB_NAME=[Database Name]
        iii. DB_SCHEMA=[Database Schema/Origin]
    b. rxconfig/Installer/rxrepository.properties
        i. DB_SERVER=[Database Server Name or IP Address]
        ii. DB_NAME=[Database Name]
        iii. DB_SCHEMA=[Database Schema/Origin]
    c. rxconfig/Server/server.properties
        i. serverName=[Database Server Name or IP Address]
        ii. databaseName=[Database Name]
        iii. schemaName=[Database Schema/Origin]
    d. rxconfig/Server.config.xml

```
<PSXBackEndCredential id="0">

      <alias>Repository Alias</alias>

      <comment/>

      <driver>Database Driver Name</driver>

      <server>Database Server Name or IP
Address</server>

      <userId>Database User Name</userId>

      <password encrypted="yes">Encrypted
Password</password>

</PSXBackEndCredential>
```

    e. rxconfig/Server/ContentEditors/ContentEditorSystemDef.xml

```
<PSXTableLocator alias="">

      <PSXBackEndCredential id="0">

            <alias>systemCredential</alias>
            <driver>Database Driver Name</driver>
            <server>Database Server Name or IP
      Address</server>
            <userId>Database User Name</userId>
            <password encrypted="yes">Encrypted
      Password</password>
      </PSXBackEndCredential>
      <Database>Database Name</Database>
      <Origin>Database Schema/Origin</Origin>
</PSXTableLocator>
```

4. Run the setDatabase.bat or setDatabase.sh script from the Rhythmyx root directory. This script will modify all application files to point to the new repository created and configured in steps 1 and 3. Open a command prompt to run this from rather than double-clicking, so if necessary you can send the output to Percussion Software Technical Support for troubleshooting.
    a. If you do not possess this utility, contact Percussion Software Technical Support and request the following files:
        i. setDatabase.bat or setDatabase.sh
        ii. KAM.jar
5. Run an installation of the **SAME VERSION AND BUILD** of Rhythmyx 4.5 (NOT Rhythmyx 5.7). Follow the instructions in *Installing Rhythmyx*. **DO NOT SELECT THE REPOSITORY INSTALLATION** (this installation will remove any customizations to system tables)**.** The Rhythmyx server installation will perform the following:
    a. Re-create the .exe files to reflect the new location
    b. Create a Server/Daemon
        i. Solaris only: optionally run the daemon installer after the upgrade installation. Remember that only builds after 20030101 support multiple Rhythmyx Daemons on the same Solaris box.
    c. Configure the current user's profile if Solaris.
6. Apply any necessary system level customizations
    a. If you accidentally ran the repository installation during step 5, re-apply customizations to system tables as well.
7. To avoid any port conflicts on the server edit rxconfig/Server/server.properties to use a unique port
    a. bindPort=9992
8. Consider disabling notifications – Notifications configuration after creating a production server replica will use the same SMTP host and Mail domain as the production server. Also, all CC lists still contain users from the production environment. To avoid confusion, make sure that production users will not receive any notifications generated during the upgrade process. The simplest solution is to simply disable notifications. Optionally, edit each workflow and each transition notification CC list to limit notifications. Editing Role membership (step 12) will also control which users receive notifications.

   a. To disable notifications, simply edit rxconfig/Workflow/rxworkflow.properties and remove the SMPT server and mail domain information to prevent the workflow engine from sending notifications (this will cause notification failure messages to appear in the server console).
      i. SMTP_HOST
      ii. MAIL_DOMAIN
   b. Another option is to set up a test SMTP server to handle notification testing. Apache Software Foundation's James mail server is available under the Apache license, and relatively simple to configure. If you run this on the same machine as the Rhythmyx server, both SMTP_HOST and MAIL_DOMAIN can be set to 'localhost'. Note that accounts must be created for each user used to test notifications on the test mail server.
      i. http://james.apache.org/

9. Make any required classpath changes for custom JAR files using one of the following methods
   a. Edit the Registry
      i. HKEY_LOCAL_MACHINE/SYSTEM/CurrentControlSet/Services/[Rhythmyx Service Name]/Parameters/CLASSPATH
   b. Create a RhythmyxServer.cp1 file
      i. A simple text file with classpath additions inserted into it. The .cpx file(s) must exist in the Rhythmyx root directory and will work on both Windows and Unix
10. Start up the replica server
    a. Make sure that no errors occur during initialization
11. Edit the Publishing Information to prevent accidental publishing to the production environment
    a. Delete all Production Publishers and Sites
    b. Create a test web server to test publishing against
       i. Do not forget any static files that the test website requires to emulate the production web server.
    c. Configure a Publisher and Site within Rhythmyx for the Upgrade Development Server to test publishing with.
12. Edit Security Information – Business users of the production CMS should not be allowed to authenticate against the supporting development and integration servers. Also, the production authentication method may differ than the development model (e.g. LDAP vs. Backend Table). Make the necessary modifications to security for these considerations.
    a. Use the Rhythmyx Server Administrator to limit the users in each Role.
    b. If notifications have not been disabled, the Role Membership will determine notification recipient lists. Remove any production users from these Roles to prevent them from receiving notifications.
13. Test the replica server to make sure everything functions identically to the Production server
    a. Test Content Editors and Content Assemblers
       i. Create a new content item with each content editor

ii. Preview the content with each variant
iii. Add related content to each slot
iv. Preview variants with related content
v. Pass 1 content item through each workflow (make sure that each content editor uses the correct workflow).

b. Test Publishing

i. Make sure no publishing configurations point to the production site or publisher.
ii. Test the published content

c. Test any other features unique to the implementation

Keep a Detailed Log

Maintain detailed development logs to facilitate planning phases of future projects. The Upgrade Checklist created during the planning phase provides a great starting point for this log. Keep maintaining this log for any future post-upgrade development work.

Run the Upgrade Installation

After setting up the replica production server and performing some QA testing on it to make sure it operates properly, move on to begin the upgrade installation of the development server. Follow the upgrade instructions in *Installing Rhythmyx 5.7*.

Perform any system modifications determined necessary

Use the upgrade path determined during the planning phase. Carefully document the steps used to apply any system modifications for use on the integration and production servers. Multi-Server Manager will not install system level changes between servers, so these processes require good documentation in order to perform them without error against the production server.

Migrate 4.5 Rhythmyx Elements to 5.7 Specifications

Perform the migration plans outlined for each impacted implementation specific Rhythmyx element outlined in the upgrade checklist.

Quality Assurance Testing

Unit tests provide the foundation of any QA plan. Each Rhythmyx element migrated to Rhythmyx 5.7 requires unit testing before installing to the integration server through Multi-Server Manager. Use the test case developed for each upgrade impacted Rhythmyx element after completing all upgrade and application migration work.

Create MSM Archives

Using the Rhythmyx Multi Server Manager 5.7.pdf and the Multi-Server Manager archive creation plan, build the archives for later installation to the integration server. Testing of the archives takes place on the integration server, but may require re-creation of the archives on the development tier. This process ensures that well-tested archives are installed to the production server.

Maintaining the current Production environment during the Upgrade process

Because the upgrade process can take weeks to complete, there is a good chance that maintenance work may need to be performed against the current Rhythmyx 4.5 production environment. This work may include (but is not limited to):

- Minor formatting changes to existing variants
- Minor workflow modifications
- Modifications to Publishing configuration

Percussion Software recommends that no major development work be performed against the current production server during the upgrade process, as this work will need to be performed twice (once in the current environment, and again in the Rhythmyx 5.7 environment). To minimize duplication of work and the risk of introducing inconsistencies between Rhythmyx servers, all major development work should be completed prior to starting the upgrade process.

If the Rhythmyx 5.7 environment is using the same servers as the current supporting environment for the production server, follow the following steps to use the old Rhythmyx environment

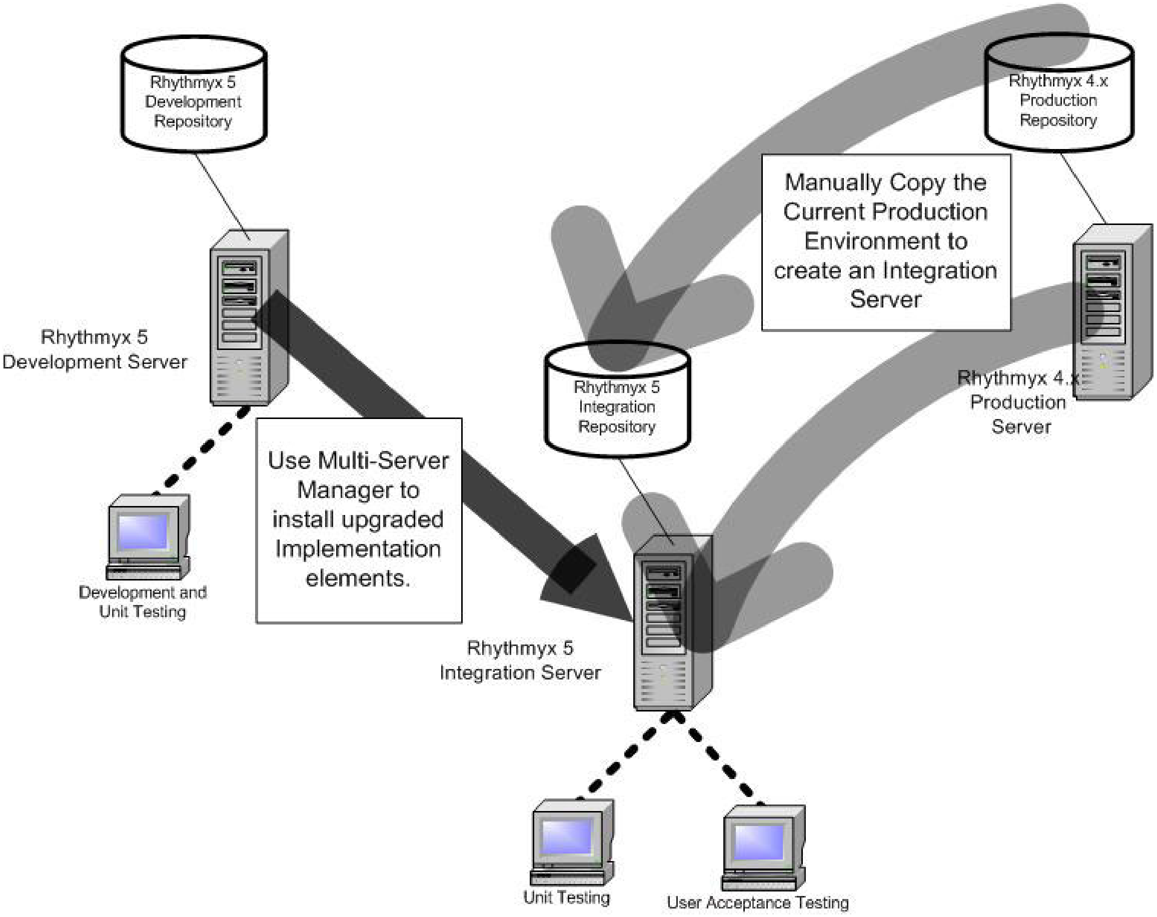
1. Shut down the Rhythmyx 5.7 upgrade server.
2. Rename the Rhythmyx directory (e.g. Rhythmyx-Upgrade)
3. Restore the name ‘Rhythmyx’ to the old Rhythmyx directory
4. Start the Rhythmyx 4.5 server

When development work has finished and changes have been applied to the production server, shut down the Rhythmyx 4.5 environment and switch back to the upgrade environment. Apply all changes applied to the production server to the upgrade server. This process will ensure that when the production server is finally upgraded, no recent changes introduced through regular maintenance are lost.

Upgrade Integration Phase

The Upgrade Integration Phase involves creating an integration server that matches all functional aspects of the Integration Tier discussed in *Modeling a Content Management System 5.7.pdf*. Make a note of how licensing can affect the use of the integration server, especially around user acceptance testing.

Figure 2 illustrates the Rhythmyx upgrade environment and activities during the Upgrade Integration Phase


Rhythmyx 5 Development Repository
Rhythmyx 4.x Production Repository
Manually Copy the Current Production Environment to create an Integration Server
Rhythmyx 5 Development Server
Rhythmyx 5 Integration Repository
Rhythmyx 4.x Production Server
Use Multi-Server Manager to install upgraded Implementation elements.
Development and Unit Testing
Rhythmyx 5 Integration Server
Unit Testing
User Acceptance Testing

Create a Production Replica

Follow the steps outlined in the Upgrade Development Phase to create a production replica to serve as the Upgrade Integration Server. This will involve creating a Publishing environment for the Integration Environment.

Keep a Detailed Log

The log started during the Development phase will become extremely useful during the setup of the Integration server. The log maintained during the integration phase will be helpful for estimating the time required for server downtime of the production environment.

Run the Upgrade Installation

After setting up the replica production server and performing some QA testing on it to make sure it operates properly, move on to begin the upgrade installation of the integration server. Follow the upgrade instructions in *Installing Rhythmyx 5.*

Apply System Modifications

Using the steps outlined while upgrading the development server, apply the system modifications to the integration server.

Backup the Upgrade Integration Server

Back up the Upgrade Installation Server file tree and repositories before installing the Multi Server Manager Archives. This backup will allow for optional practice of the Archive installation process a second time before moving to the Production Server.

Install Multi-Server Manager Archives created during the Development Phase

Using the *Rhythmyx Multi Server Manager 5.7.pdf* and Multi-Server Manager archive creation plan, install the archives created on the Development server to the Integration server. When mapping IDs between servers, remember that any existing Rhythmyx elements will retain the same ID between servers, but any new Rhythmyx elements added during the upgrade project receive a unique ID on the Integration Server during archive installation. This ID may or may not match identically to the Development server, but Multi-Server Manager will handle the ID Mapping between servers.

Develop an installation procedure to use on the Production Server. Optionally restore the integration server from the backup and practice this procedure before installing these archives to the production server.

Quality Assurance Testing

1. Test all Migrated Rhythmyx elements
    a. Use the Test Case designed for each
2. Test Content Editors and Content Assemblers
    a. Create a new content item with each content editor
    b. Preview the content with each variant
    c. Add related content to each slot
    d. Preview variants with related content
    e. Pass 1 content item through each workflow (make sure that each content editor uses the correct workflow).
3. Test Publishing
    a. Use a test site to test publishing.
    b. Test the published content
4. Test any other features unique to the implementation

User Acceptance Testing

A basic business requirement is for users to perform the same day-to-day CMS tasks in the upgraded environment that they can today. User Acceptance Testing is intended to uncover issues in the implementation using a test population. This testing gives developers a chance to address the issues before they affect the entire user base.

Design use cases that will help the testers test the upgraded CMS. Some examples:

- Follow standard content items from start to finish through all stages of the workflow
- Use Content Explorer to search for content
- Edit currently public content and publish the changes out to the site

Next, select a core group of end users to perform acceptance testing on the upgraded integration server. At this point, modify the security settings of the Integration server, and add these users into the appropriate Roles. Optionally, enable notifications but change each message subject and/or body to indicate User Acceptance Testing to prevent confusion among testers. Be sure to remove any production user's email addresses from the CC lists before enabling notifications. All testers may still receive notifications from the current production environment regarding content assigned to them. Inform all testers about the notifications.

Develop End User Documentation

Feedback obtained during UAT will help decide which areas will require documentation. Highlight all Rhythmyx 5.7 features that change the existing process in any way (business logic, GUI), with helpful pointers to the installed Rhythmyx End-User documentation that describes the feature in Rhythmyx 5.7. Any process changes (e.g. workflow, folders, communities, locales) will require more in depth documentation, as the specific implementation will enforce business logic outside the scope of the standard Rhythmyx End-User documentation.

Develop User Training

Structure End User Training around the new Content Explorer interface. Walk the users through the new features and UI changes. Also explain any process changes. Questions raised during UAT indicate possible areas for training in determining topics for User Training.

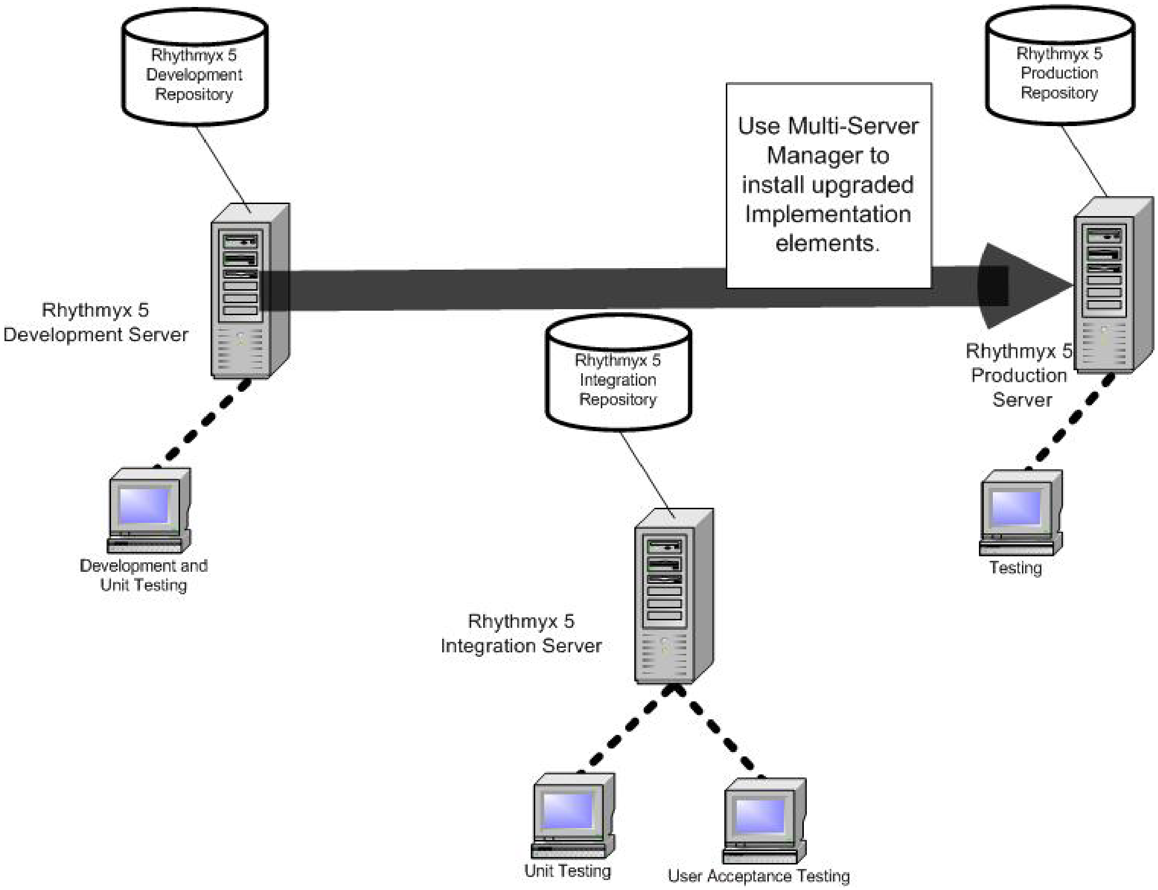
Perform Pre-Production User Training

Exposing more users to Rhythmyx 5.7 prior to upgrading the production server will ease the transition for the users and give developers more feedback earlier in the process.

Upgrade Production Phase

This stage of the upgrade project will involve upgrading the current Production Server, which maps identically to the Production Tier discussed in *Modeling a Content Management System 5.7.pdf*.

Figure 3 illustrates the Rhythmyx upgrade environment and activities during the Upgrade Production Phase


Rhythmyx 5 Development Repository
Rhythmyx 5 Production Repository
Use Multi-Server Manager to install upgraded Implementation elements.
Rhythmyx 5 Development Server
Rhythmyx 5 Integration Repository
Rhythmyx 5 Production Server
Development and Unit Testing
Testing
Rhythmyx 5 Integration Server
Unit Testing
User Acceptance Testing

Backup the Production Server

Make a backup of both the Production Rhythmyx Server file tree and repositories before proceeding any further.

Schedule Production Server Downtime

Base estimates of server downtime on the time required to:

- perform the upgrade installation,
- install the Multi Server Manager archives
- run the Inline Link Converter to upgrade content
- Perform QA testing.

Create estimates based on the amount of time these tasks took on the integration server. If possible, select a time with the end users will be least impacted (evening, weekend, etc…).

If Production Server downtime is not an option, then the upgrade must be performed with the hot-swap server first. If the hot-swap server is sharing the same repository as the production Rhythmyx server, create a replica of the production repository and repoint the hot-swap server to the replica (refer to steps 3-4 outlined in the Upgrade Development Phase). Perform the upgrade steps against the hot-swap server, then switch it out with the production server. Create a new-hot-swap server based on the Rhythmyx 5 production server (former hot-swap server). All content contributed during the upgrade process will need to be re-created on the upgraded production server. Contact Percussion Technical Support if you need any assistance planning a hot-swap upgrade.

Bring the Production Server offline

The Rhythmyx Upgrade work will require that the server be running for the Multi-Server Manager archive installation. Before proceeding, make sure that end users access to the Rhythmyx Server is disabled.

1. Take the Production Server off the physical network (e.g. unplug the Ethernet cable).
2. Only allow the range of IP addresses belonging to the Rhythmyx Development group to connect to the Production Server during the upgrade process.

Run the Upgrade Installation

Take the server offline to prevent users access until restoring stability. Note that the Multi-Server Manager client must still be able to connect to this server. If running on Windows, Multi-Server Manager can run on the same machine if Developer Tools have been installed. If running on Solaris, a windows client must be able to connect. Consider changing the Rhythmyx port during this process. Follow the upgrade instructions in *Installing Rhythmyx 5.7.*

Apply System Modifications

Using the steps outlined while upgrading the development server, apply the system modifications to the production tier.

Install Multi Server Manager Archives created during the Development Phase and tested during the Integration Phase

Install the Multi-Server Manager Archives using the installation plan developed during the Integration Phase.

Quality Assurance Testing

1. Test all Migrated Rhythmyx elements
    a. Use the Test Case designed for each
2. Test Content Editors and Content Assemblers
    a. Create a new content item with each content editor
    b. Preview the content with each variant
    c. Add related content to each slot
    d. Preview variants with related content
    e. Pass 1 content item through each workflow (make sure that each content editor uses the correct workflow).

3. Test Publishing
    a. Use a test site to test publishing.
    b. Test the published content
    c. Test any other features unique to the implementation

Distribute User Documentation

Send out the End User Documentation to all end users.

Perform User Training

Provide any required training sessions for untrained users.

Post Upgrade Phase

Deprecate Old Rhythmyx Environment

The Rhythmyx servers that supported the old Rhythmyx 4.x environment can now be archived and deprecated.

Use the new Multi-Tiered Development Practices

Moving forward, all maintenance and new development work should follow the Multi-Tiered Development guidelines outlined in the *Modeling a Content Management System 5.7.pdf*.

Plan any new Rhythmyx 5 development work

Now that the Rhythmyx environment is fully upgraded, consider developing any enhancements to the current implementation based on new features available in Rhythmyx Version 5, 5.5, 5.6, or 5.7.

System Modifications Checklist

The System Modifications checklist delineates modifications to the Rhythmyx server in Version 5 that either affect system stability of a system upgrade from version 4.5 or may cause the upgraded system to fail. Some items in the System Modifications checklist affect most, if not all Rhythmyx implementations. These items are formatted in **bold text**.

Following the checklist is a section describing these changes in more detail.

System Modifications

Operation Issues

System Applications

System Tables

JUMPSTART deprecated

RXRELATEDCONTENT removed

System Table modifications

System Extensions

Workbench

Publisher

XML Loader

Rhythmyx Daemon

Implementation Issues

Sample Rhythmyx Applications

Content Editor Definition XML Files

- Shared Definition Files
- Local Definition Files

Custom Controls

- eWebEditPro
- Microsoft Word Connector

Customized Extensions

PSCToolkit

Content Assemblers and Custom Stylesheets

- Internationalization
- Custom Stylesheets

Custom SQL Queries

Custom Database Views

Custom Searches

Custom Components and Visibility Rights

Purge Content Rights

Custom Banners

Customized Cascading Stylesheets and General System Look and Feel

Automated Indices

Communities

Workflows

Multi-Server Manager

Restoring System Stability: System Modifications

This section provides detailed information about changes to Rhythmyx that either affect system stability of a system upgrade from version 4.5 or may cause the upgraded system to fail.

System Applications

The upgrade process always overwrites system applications. You can easily recognize system applications by their "sys_" prefix. Because system applications are overwritten on all upgrades, Best Practice is not to modify any of these applications. If you have modified system applications, your changes **will be lost**. Before upgrading, note any modifications you have made so you can implement them again after upgrade. Also, check with Percussion Software Technical Support to find out whether any new features added that make your modifications unnecessary.

System Tables

JUMPSTART Table Deprecated

In Rhythmyx 5.5, the JUMPSTART table was deprecated. It is no longer necessary. The upgrade process does not remove this table, however, because customer systems may query it and it may contain vital data.

RXRELATEDCONTENT Table Deprecated

In Rhythmyx 5.7, the Relationship Engine extends the ability to associated objects in Rhythmyx, and related content is one example of such a Relationship (the Active Assembly Relationship). The Relationship Engine tables thus supercede the RXRELATEDCONTENT table. The installation process removes the table, but adds a database view for backwards compatibility in systems that queried it.

Customized System Tables

If you have customized system tables (such as adding a new column to a system table), upgrade of the Repository may fail. If you have modified any system tables, check the `tablefactory.log` file for errors after you upgrade.

System Extensions

New optional parameters have been added to some system extensions. The Workbench notifies you when you open an application that includes one of these new parameters.

Workbench

A Rhythmyx version 4.5 Workbench cannot connect to a Rhythmyx 5.7 server. You should upgrade all Developer Tools installations when you upgrade the Rhythmyx server.

The Rhythmyx 5.7 Workbench is backwards compatible with Rhythmyx 4.5 servers.

Publisher

In Windows, the Rhythmyx publisher service is renamed to Percussion Rhythmyx Application Server. All Publisher Hubs in Windows environments must also be upgraded.

XML Loader

The XML Loader is no longer supported. Use the Enterprise Content Connector to upload syndicated XML data.

Rhythmyx Daemon (Solaris only)

In builds prior to January 2003, the Rhythmyx daemon ran as root by default. It was possible to modify the Rhythmyx daemon to run as a different user, but doing so required a manual cleanup process because the root user owned files created by the Rhythmyx server. To modify the daemon to run as a different user, you had to change the ownership of these files to the Unix user under which Rhythmyx was installed. You could do this with a recursive chown command by the root user:

```
#chown -R[unixuser][Rhythmyx Home Directory].
```

In January 2003, Percussion Software introduced a new Rhythmyx daemon. Installation of the daemon is still done as a post-install process by the root user running `InstallDaemon.sh`. The new daemon runs Rhythmyx as the Unix user that installed Rhythmyx, and supports multiple instances of the Rhythmyx server. Modifications to the InstallDaemon.sh script clean up any file ownership problems.

Previous versions of the Rhythmyx daemon added a line to `/etc/inittab`. The system administrator should remove this line manually. (The file `/etc/inittab` is a system file that controls the server initialization process. You should back it up before making any changes.) To modify the file:

1. Back up the `inittab` file: `#cp/etc/inittab /etc/inittab.bak`
2. Open `/etc/inittab` in a text editor.
3. Delete the following line: `rh:2:once:/etc/rc2.d/S15RhythmyxD`
4. Verify your changes and save the file.

Restoring System Stability: Implementation Issues

Sample Rhythmyx Applications

The installer does not upgrade sample applications, which typically begin with the prefixes “rx_” and “cas”. If you want to upgrade sample applications, delete them before upgrading. The installer will install the new version of these applications.

Content Editor Definition XML Files

The <PSXField> element has two new attributes to facilitate searches.

- defaultSearchLabel
  If a field does not have a UIDef, or if the UIDef is overridden in a Content Editor, Rhythmyx uses the value of this attribute as the Label for the field in any search dialog it displays to the user. If both the defaultSearchLabel attribute and the UIDef are missing, Rhythmyx displays the field name.
- userSearchable
  This optional attribute specifies whether to make this field available to users when they define a search. If this attribute is not included, Rhythmyx assumes a default value of “yes”.

For details about these attributes, see the comments in the `Rhythmyx/DTD/sys_BasicObjects.dtd`.

Shared Definition Files

The installer does not upgrade existing shared definition files. If you want to add the new <PSXField> attributes (see above) to a shared definition file, you must add it manually.

The shared definition file `sys_JumpstartSharedDef.xml` has been deprecated. Applications that rely on this table will still work, but you can re-architect them in Rhythmyx 5.7 without these shared fields. Note that if you decide to modify content editors to exclude these fields, you must also update the corresponding Content Assembler applications that select from the JUMPSTART table as well. (See above, "JUMPSTART Table Deprecated" on page 28).

Local Definition Files

Two new attributes have been added to the <PSXContentEditor> element,

- enableRelatedContent
  This attribute specifies whether the Content Type can have dependents in Active Assembly Relationships. The default value is "yes".
- objectType
  This attribute is used in Relationship processing.

Thus, a typical <PSXContentEditor> element in Rhythmyx 4.5:

```
<PSXContentEditor contentType="1" workflowId="1">
```

in Rhythmyx 5.7 becomes:

```
<PSXContentEditor contentType="1"
enableRelatedContent="yes" objectType="1"
workflowId="1">
```

The following children have been removed from the <PSXUrlRequest> child elements:

```
<PSXUrlRequest name="RelatedLookupURL">

<PSXUrlRequest name="VariantListURL">

<PSXUrlRequest name="ContentSlotLookupURL">
```

The Relationship Engine makes these URL requests unnecessary. The SharedFieldInclude of the Related Content is also no longer necessary:

```
<SharedFieldIncludes>

  <SharedFieldGroupName>relatedcontent</SharedFieldGro
  puName>

</SharedFieldIncludes>
```

If you implemented any shared field excludes of fields in the `relatedcontent` shared group, you must also remove these exclude statements as well.

Custom Controls

EditLive! for Java

In Rhythmys 5.7, a new rich text DHTML editor is provided. For details, see "Ephox EditLive! for Java DHTML Editor" on page 49.

Microsoft Word Connector

Minor enhancements were added to the Word Connector in several builds of Rhythmyx 4.5. If you have a build prior to 20030109, you may want to check your implementation of the Word Connector for compatibility and upgrade to the latest features. Builds subsequent to 20030109 include minor bug fixes, but no new functionality.

Customized Extensions

In Rhythmyx 5.5, the XML parser was changed from xml4j to Xerces. Xerces enforces XML rules more strictly than xml4j. You may need to modify XSL files that support extensions to ensure that they conform to the stricter rules. For more information about modifying XSL files, see "Content Editors and Custom Stylesheets" below. For example, the following statement worked in Rhythmyx 4.5:

```
parentDocElement.appendChild(
childContainerElement.cloneNode(true) );
```

The same statement would not work correctly in Rhythmyx 5.5 and later. You would have to substitute the following statement:

```
parentDocElement.appendChild(
parentDoc.importNode(childContainerElement,true) );
```

Xerces will not clone an element then attach it to a new document. You must import the element.

Any custom extension that depends on a system application should be tested to ensure that changes to the underlying tables or XML structure of the application have not rendered the application incompatible with the extension. In most cases, more information is not available, and you may be able to modify your extension to take advantage of additional XML nodes.

In all cases, you can use application tracing to debug custom extensions.

PSCToolkit

Several extensions from the PSCToolkit have been integrated into Rhythmyx.

- PSIncrementalContentFilter (now sys_IncrementalContentFilter)
- PSAutoRelatedContent (now sys_AutoRelatedContent)
- PSAddAssemblerInfo (now sys_AddAssemblerInfo)
- PSAddChildInfo (note sys_AddChildInfo)

You should update any calls to these extensions to use the version of the extension that has been integrated into Rhythmyx.

Other extensions from the PSCToolkit have been updated for Rhythmyx 5. Customers should replace their existing PSCToolkit with the Rhythmyx 5 PSCToolkit. If you have created custom extensions based on extensions in the PSXToolkit, you should compare them to the new version in the Rhythmyx 5.7 PSCToolkit, and, if necessary, update and recompile the customized extensions.

The functionality of one exit, PswfCommunityCheck, has been integrated into the standard functionality of Rhythmyx. Implementations that use this exit no loner need it and can remove it from Content Editor resources to which it has been attached.

Content Assemblers and Custom Stylesheets

XSL Version Upgrade

The Rhythmyx XML processor has been changed from xml4j to Xerces. Xerces supports XSL 1.1 functionality. The installer should upgrade all XSL headers, but you may need to resplit HTML for stylesheets without Slots to upgrade the headers.

The xml4j processor allowed an XSL Version 1.0 stylesheet to use XSL 1.1 functionality. Xerces, however, does not. You will need to update the stylesheets to version 1.1 to avoid errors.

- Example 4.5 stylesheet:

  ```
  <xsl:stylesheet version="1.0"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  xmlns:saxon="http://icl.com/saxon" extension-
  element-prefixes="saxon">
  ```

- Example 5.7 stylesheet:

  ```
  <xsl:stylesheet exclude-result-prefixes="psxi18n"
  extension-element-prefixes="saxon" version="1.1"
  xmlns:psxi18n="urn:www.percussion.com/i18n"
  xmlns:saxon="http://icl.com/saxon"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
  ```

If you do not upgrade your stylesheets, you may encounter errors similar to the following:

```
An exception occurred while processing the Article XML
document through the file:casArticle/Article.xsl style
sheet: Error while transforming the XML file with
stylesheet.To use a result tree fragment in a path
expression, either use exslt:node-set() or specify
version='1.1'; SystemID: file:casArticle/Article.xsl;
Line#: -1; Column#: -1
```

Internationalization

Two new optional attributes have been added to Rhythmyx XSL stylesheets to support internationalization.

- exclude-result-prefixes=”psxi18n”
- xmlns:psxi18n=”urn:www.percussion.com/i18n”

Example:

```
<xsl:stylesheet exclude-result-prefixes="psxi18n"
extension-element-prefixes="saxon" version="1.1"
xmlns:psxi18n="urn:www.percussion.com/i18n"
xmlns:saxon="http://icl.com/saxon"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
```

Rhythmyx adds these new attributes to any stylesheet split in a Rhythmyx 5.7 Workbench or Rhythmyx 5.7 XSpLit.

For more details about internationalization, see Internationalizing and Localizing Rhythmyx.pdf.

Custom Stylesheets

You must upgrade all custom stylesheets to match the standards delineated in the previous sections. Check the following locations for custom stylesheets:

- Extensions that use stylesheets
- rx_resources\stylesheets\assemblers\rx_Globals.xsl
- rx_resources\stylesheets\assemblers\rx_InlineLinks.xsl
- rx_resources\stylesheets\assemblers\rx_popmenu.xsl
- rx_resources\stylesheets\assemblers\rx_Slots.xsl
- rx_resources\stylesheets\rx_Templates.xsl
- rx_resources\stylesheets\ (any XSL file)

You should also test any custom stylesheets that depend on system-generated XML against the upgraded server. Changes to XML structures may cause incorrect processing of these XSLs. Use application tracing to help with debugging.

Custom SQL Queries

Queries that depend on system tables to which new columns have been added may fail, depending on the syntax of the SQL.

Custom Database Views

Review database views that rely on system tables and make any necessary updates to the SQL statements in the views.

Custom Components and Visibility Rights

Content Explorer does not use the custom components developed for Content Administrator.

Use the following options to control visibility rights for business users:

- for Folders, use Folder security
- for Views and Searches use the View/Search Editor
- for Actions and Action Menus, use the Action Menu Editor.

Custom Searches and Views

Custom searches and Views are now integrated into the Content Explorer. You must re-create your custom searches in the Rhythmyx Workbench using the View/Search Editor. Custom searches developed for the Content Administrator will no longer work.

You can use custom search resources you developed for the Content Administrator as the basis for custom searches implemented through the Workbench, but unless the search involves a complex query (a query that selects data from one table based on matching data in a second table), it will probably be more efficient to use the functionality available in the Workbench.

Searches and Views in Rhythmyx 5 return a list of content items. The Content Explorer applet then applies a Display Format against the list of content items. The Display Format formats information about each item. Most custom Searches and Views migrated from Rhythmyx 4.5 will require a display format as well. You may choose to use one Display Format across all Views and Searches.

If you do decide to base a search on an existing query, be sure to check for system changes that may cause the query to fail. Search and View query resources must conform to the sys_cxView/view.dtd.

NOTE: Currently, when Content Explorer runs on a portlet, it has difficulty creating the new applet windows required to implement the Rhythmyx 5 search.

Purge Content Rights

Purge is now an Action, and you set its visibility in its Menu Entry in the Action Menu Editor.

During upgrade, the installer will transfer purge rights from the ca_nav component to the Purge Action Menu Entry. Use the Action Menu Editor to change the configuration of these rights.

Custom Banners

Banners customized in Rhythmyx 4.5 will not function properly in Rhythmyx 5.7 due to the following changes to Login Details

- The Login Details now includes Locale.
- Login Details no longer expands because it is no longer an HTML table. (NOTE: Currently the image used as a table background for Login Details does not render consistently across all browsers. This is a known issue.)

NOTE: Another reason that banners customized in Rhythmyx 4.5 will function incorrectly in Rhythmyx 5.7 is a bug in the Rhythmyx 5.7 banner that prevents hiding tabs from non-authorized users. This is a known issue that will be fixed in a future version of Rhythmyx.

If you customized the banner in Rhythmyx 4.5, you should change the registration of the banner component (cmp_banner) to point at the default banner (`../cmp_banner/banner.html`). If you still want to use a customized banner, you should develop a new custom banner starting with the Rhythmyx 5.7 banner.

Customized Cascading Stylesheets and General System Look and Feel

The installer overwrites the cascading stylesheets (CSS) used in rendering the Content Administrator when upgrading to Content Explorer. Content Explorer gives individual users the power to modify their own interface, so it is not necessary to repeat customization of these stylesheets. However, you can customize these stylesheets if you want to change the default look and feel of Content Explorer.

Automated Indices

Implementation of automated indices has changed. The addition of the sys_casAutoRelatedContent exit (originally part of the PSXToolkit, but now incorporated as a system exit) mitigates the need to manipulate the $related XSL variable. The exit now handles this automatically.

Automated indices developed for Rhythmyx 4.5 will still function correctly. You should still check these applications, however, to ensure that changes to sys_addAssemblerInfo DTD does not cause them to function incorrectly.

Inline Links and Inline Images

Inline Links and Inline Images have been converted to Active Assembly Relationships in Rhythmyx 5.7. The installer does not upgrade existing Inline Links and Inline Images because the Active Assembly Relationships require Slots and Variants, which are customized for each implementation. Existing Inline Links and Inline Images will work correctly during Publishing, but you will not be able to monitor them using Impact Analysis until they are upgraded. For details about upgrading Inline Links and Inline Images, see "Related Content, Inline Links, and Inline Images" on page 38.

Communities

In Rhythmyx 4.5, Communities were an optional feature. In Rhythmyx 5.7, Communities are mandatory. Before upgrading, review the documentation on Communities. (See "Modeling Communities" in Modeling a Rhythmyx Content Management System 5.7.pdf and "Implementing Communities" in Implementing the Rhythmyx Business User's Interface 5.7.pdf. Both of these .pdf files are available on the Rhythmyx 5 CD-ROM.)

The following Rhythmyx 5.7 features rely on Communities:

- Folder Security
- Action Menu visibility
- View and Search visibility

A Site must be assigned to a Community for members of that Community to be able to manage it using Site Explorer.

Community Roles, Workflow Roles, and Parallel Workflows

Rhythmyx 5.5 added a distinction between Community Roles and Workflow Roles. Rhythmyx can use either Role to log into Rhythmyx, but once in the system, the Roles are used very differently. For details, see "Parallel Workflows for Multiple Communities" on page 40.

Workflows

Transition Enhancements

Rhythmyx 5.7 provides more options for specifying how Transitions are approved. You can now specify that users in specific Roles must approve Content Items before they make a Transition. For details, see "Transition Roles" on page 42.

An option is now available to assign a Transition as the Default Transition from a State. This option is required to make Strong Dependencies work. You must assign Default Transitions manually, however. The installer does not define default Transitions during upgrade.

State Enhancements

A new option has been added to the Publishable option for States. The options available now are

- Publish
- Unpublish
- Ignore (When processing this option, the Publisher Publishes the item if it is already Public, or leaves it unpublished if it is not already Public. This option is used with the Quick Edit feature; see "Implementing Quick Edit" on page 42.)

State Role Enhancements

A new option has been added to State Roles called "Show in Inbox". This property takes a boolean (yes/no) value. This property is used to limit the amount of content returned in the user's Inbox view. In particular, it is used to keep Public and Archived Content Items out of the Inbox View.

All upgraded Workflows default each State Assigned Role a value of "y" (yes) for this property, which displays Content Items in that State in the user's Inbox. Change the value to "n" (no) for the State Roles if you want to keep content from that State out of the user's Inbox View.

The behavior of the Assignment Type *None* has changed. In Version 4.5, if a Role had the Assignment Type *None* for a State, Rhythmyx displayed Content Items in that State to Members of the Role, but did not allow the Members to preview or edit the items. In Version 5.7, Rhythmyx does not display Content Items to Members of Roles with the Assignment Type *None* for a State.

Workflow Role Issues

In all versions of Rhythmyx, each Workflow should include an Administrator Role. Users in this Role have special rights (such as force check-in) to prevent isolation Content Items in the Workflow. The value entered as Administrator must make a case-sensitive match to an existing Role in the system to make these rights available. If no match is found, no user has administrative rights in the Workflow. Thus, if you have a

Role defined as “Administrator” (note capitalization) in your system, but you enter “administrator” (all lowercase) as the Workflow Administrator, Rhythmyx does not find a match, and no user has administrative rights.

When assigning Roles to a State in Rhythmyx 4.5, you had to enter the name of the Role manually. In Rhythmyx 5.7, a drop list control has been substituted for the text box control. State Assigned Roles are case-sensitive, so you should check existing State Role assignments to be sure that you have entered them correctly.

Multi-Server Manager

Archives created under Rhythmyx 4.5 are not valid for installation on Rhythmyx 5.7 servers. You will need to re-create these archives under Rhythmyx 5.7. When you begin this process, Multi-Server Manager will notify you that changes to the server affect the Descriptor, and will ask you to review the Descriptor and update it to accommodate those changes.

Percussion Software recommends that you complete your upgrade (including all changes required to restore your system to stability) before re-creating your archives.

New Features in Rhythmyx 5.0, 5.5, 5.6 and 5.7

This section contains two subsections. The first subsection delineates new features added to Rhythmyx 5.0 and implementation details for these features. The second subsection delineates new features added in Rhythmyx 5.5 and implementation details for those features.

New Features added in Rhythmyx 5.0

Views and Searches

Related Content, Inline Links, and Inline Images

Community Enhancements

Quick Edit

Web Server Security Provider and Single Sign-on

Upgrading to Rhythmyx 5.0 Features

Rhythmyx Version 5.0 introduced a variety of new features. Implementer intervention is required to gain the full benefit of the following new features:

- Views and Searches in Content Explorer
- Related Content
- Parallel Workflows for Multiple Communities
- Web Server Security Provider and Single Sign-on

Views and Searches in Content Explorer

In previous versions of Rhythmyx, creating a custom View or Search Box required cloning and customizing Rhythmyx applications. These applications were integrated into the Content Administrator interface as Components.

These customized Views and Search Boxes will no longer work in Content Explorer. You must recreate them, but the process has been significantly simplified in Rhythmyx Version 5.5.

For details about creating a custom View, see "Maintaining Content Views" in Implementing the Rhythmyx Business User Interface, or in the Rhythmyx Workbench online Help.

Related Content, Inline Links, and Inline Images

The only association between Content Items recognized by previous versions of Rhythmyx was related content. Related content associations were stored in a special table dedicated to that purpose (RXRELATEDCONTENT).

Rhythmyx Version 5.0 introduced an extended Relationship engine that offers much more flexibility. The new Active Assembly Relationship replaces the former related content association. The Rhythmyx installer automatically converts related content associations to Active Assembly Relationships when upgrading the Rhythmyx Repository.

Inline images and inline links are also implemented as Active Assembly Relationships beginning in Rhythmyx Version 5.0. Because all Active Assembly Relationships, including inline links and inline images, require the use of Slots and Variants, the server does not automatically upgrade existing inline links and inline images. First, you must define your inline Slots and inline Variants, then use the upgrade application, PSInlineLinkConverter to upgrade inline images and inline links to Active Assembly Relationships.

To convert existing inline links and inline images to Active Assembly Relationships:

1. For each Content Editor field that uses a rich text DHTML control (sys_eWebEditPro or sys_EditLive), check the following checkboxes on the Display Control Properties for the control:
   - mayHaveInlineLinks
   - cleanupBrokenInlineLinks
2. Develop your inline image and inline link Slots. See "Creating Custom Inline Slots" in *Implementing Rhythmyx Content Editors and Content Assemblers*.
3. Develop your inline image and inline link Variants. See "Creating Inline Image and Inline Link Variants" in *Implementing Rhythmyx Content Editors and Content Assemblers*.
4. Run the InlineLinkConverter application.

Running the InlineLinkConverter Application

Three versions of the InlineLinkConverter application are available:

- InlineLinkConverter.exe (used in Windows environments)
- InlineLinkConverter.bin (used in Unix environments)
- InlineLinkConverter.jar (can be used in either environment to run special conversions)

All three versions, as well as the mandatory properties file, are stored in the directory Rhythmyxroot/InlineLinkConverter.

Before running the InlineLinkConverter application, you must modify the InlineLinkConverter properties file, located in the InlineLinkConverter directory, specifying the user you want to use to make the conversion and the users password. (Default values are `user=admin1` and `password=demo`.)

After you update the properties file, you can run the application in the manner appropriate to your environment. The application converts all inline links and inline images, and generates a log file in the InlineLinkConverter directory.

In some cases, you may want to convert a specific Content Item, or all of the Content Items of a Content Type. In that case, you must modify the properties file. In the `contentType` property, specify the Content Type whose Content Items you want to convert (for example, `contentType=Article`). In the `contentId` property, specify the ID of the Content Item you want to convert (for example, `contentId=521`), and in the `revision` property, specify the Revision of that Content Item that you want to convert. Note that if you want to use the `contentId` property, you must specify the `contentType` and the `revision`. (If you do not specify these properties, the InlineLinkConverter cannont locate the item to convert). You must also remove comment markers on these properties.

The following code shows an example of a modified properties file:

```
#/*[ PSInlineLinkConversion
]***********************************************

# *

# * COPYRIGHT (c) 1999 - 2003 by Percussion Software, Inc.,
Stoneham, MA USA.

# * All rights reserved. This material contains unpublished,
copyrighted

# * work including confidential and proprietary information of
Percussion.

# *

#
*****************************************************************
************/

username=admin1

password=demo

webServicesLocation=http://localhost:9980/


############################

# optional properties below

############################
```

```
#
# contentType - the name of the content type to convert
#
contentType=WordArticle

#
# The following optional properties must have a contentType
# and contentId and revision specified.
#
# contentId - the specific content id
# revision  - the specific revision
#
contentId=521
revision=1
```

To run the converter in this case, start a DOS window or console, change to the <Rhythmyxroot>/InlineLinkConverter directory and enter the command:

```
..\JRE\bin\java -cp
.\InlineLinkConverter.jar;..\lib\xmlParserAPIs.jar;..\lib\xercesI
mpl.jar;..\lib\saxon.jar;..\lib\rxclient.jar;..\lib\rxmisctools.j
ar;..\lib\rxserver.jar;..\lib\soap.jar;..\lib\mail.jar;..\lib\act
ivation.jar
com.percussion.inlinelinkconverter.PSInlineLinkConverter
```

Parallel Workflows for Multiple Communities

In previous versions of Rhythmyx, Workflows could only support one Community. If additional Community wanted to use the same Workflow, implementers had to create a unique copy of that Workflow for each Community, with unique Role names for each Workflow. For example, if you had an Article Workflow you wanted to use for your Human Resources Community and your Marketing Community, you would need to create two clones of the Article Workflow.

Community	Workflow Name	Workflow States	Workflow Roles
Human Resources	HR_Article	HR_Draft HR_Edit HR_Approval HR_QA Public Archive	HR_Author HR_Editor HR_Approver HR_QA
Marketing	Mktg_Article	Mktg_Draft	Mktg_Author

Community	Workflow Name	Workflow States	Workflow Roles
		Mktg_Edit Mktg_Approval Mktg_QA Public Archive	Mktg_Editor Mktg_Approver Mktg_QA

Beginning with Rhythmyx Version 5.0, Workflows can work with multiple Communities. Thus, instead of the above example, you could have one Article Workflow:

Workflow Name	Workflow States	Workflow Roles
Article	Draft Edit Approval QA Public Archive	Author Editor Approver QA

To use parallel Workflows, each user must be a Member of at least two Roles. One Role is associated with the Community. The other Role is associated with the Workflow. When determining whether to grant a user access to a Content Item, Rhythmyx checks to see whether the use is in the Community Role of the Community associated with the item, and then whether the user is in a Workflow Role assigned to the State of the item. If Rhythmyx finds that both are true, it grants the user access to the Content Item. If either test fails, the user is not granted access.

You do not have to convert to parallel Workflows. Your existing Workflows will continue to work normally, with no loss of functionality or efficiency. Converting to parallel Workflows may simplify Workflow maintenance, but you will need to evaluate whether it is worth the investment. However, when adding additional Communities to your Rhythmyx Content Management System in the future, using parallel Workflows will be simpler to implement and more efficient to maintain.

If you decide to convert from your existing Workflow structure to parallel Workflows, decide whether you want to consolidate existing Workflows by updating one existing Workflow or by creating a new Workflow available to all Communities.

To implement parallel Workflows:

1. Create a Role for each of your Communities. Assign the users in each Community to the appropriate Role. For details about Role maintenance, see "Maintaining Roles" in the Rhythmyx Server Administrator online Help.
2. Create your Workflow Roles. Remember, you must create Workflow Roles both in the Rhythmyx Server (see "Maintaining Roles" mentioned above) and in the Workflow Administrator (see Roles in the Workflow Administrator online Help).
3. Remove Workflow Roles (for example, Admin, Editor, Author) from Community Registrations. Remove Community Roles from Workflows. Only Community Roles should be included in Community registrations and only Workflow Roles should be included in Workflows.

Transition Roles

In Rhythmyx 4.5, you could either assign one Role to a Transition, or you could make it available to all Roles. These limited options required duplication of Transitions between States when you wanted multiple Roles, but not all Roles, to make a Transition from a State.

In Rhythmyx 5.0 Role assignment to Transitions was enhanced, allowing you to assign the Transition to all Roles, or to a set of one or more Specified Roles.

Associated with this enhancement is the ability to set the number of approvals required for a Transition. You can require a specified number of approvals, each from a unique user; or you can require an approval from each Role assigned to the State.

To specify the Role options for a Transition:

1. On all Transitions for the Workflow, in the **Approval Type** drop list, choose *Each Role*.
2. If you want all Roles assigned to the State to approve the content before it can make the Transition, in the **Transition Role** drop list, choose *All Roles*.
3. If you want only specific Roles assigned to the State to approve the content before it can make the Transition:
   b. In the **Transition Role** drop list choose *Specified Roles*.
   c. Click Add Transition Role.
   d. Rhythmyx displays the Add Transition Role page.
   e. Choose the Role you want to add. Options are all Roles in the Workflow not currently assigned as Transition Roles to this Transition.
   f. Click [**Save**].
   g. Repeat steps e through h to add more Roles. Each Role you add must approve the content before it can make the Transition.

Implementing Quick Edit

In prior versions of Rhythmyx, users were not barred from checking out and editing Content Items in a Public State. While Percussion Software recommended Transitioning content to an editable State before checking it out, customers routinely checked out and edited Public content because the process of Transitioning and checking out content was cumbersome and inefficient.

In Rhythmyx Version 5, Workflow processing was tightened to prevent this practice. Users are no longer able to check out and edit content when it is in a Public State.

To make the process of accessing Public content for minor edits less onerous, new Quick Edit processing has been added to the system. Quick Edit is a option available in the Edit menu for Content Items in Public States (Edit > Quick Edit). This option Transitions the Content Item from a Public State to a Quick Edit State and checks it out to the user in one action.

Quick Edit requires some refinement of each Workflow in the system. For details about implementing Quick Edit, see “Implementing Quick Edit” in *Maintaining Workflows in Rhythmyx*.

Required Updates to Content List Resources Due to Quick Edit

In developing Quick Edit, Percussion Software has extended the definition of the Publishable property of States. A new value has been added for this property, *i* (ignore). Percussion has also made it possible to add more values to this property. For details about extending the Publishable property, see “Extending Publishable States” in *Maintaining Workflows in Rhythmyx*.

The intended behavior of Publishable = i is to ignore whether Content is Publishable or Unpublishable. If a Content Item is not yet Public, and it is in a State where Publishable is set to I, the Content will not be published. If the Content Item is already Public, it will be Published.

To make this value work correctly, existing Content Lists will have to be modified. One of the criteria used to select content for Content List resources is the value of Publishable for the State. This value is stored in STATES.CONTENTVALID. The common practice when developing Content List resources for publishing has been to select content in States where `STATES.CONTENTVALID = y`. When developing Content List resources for unpublishing, content is selected where STATES.CONTENTVALID does not equal y (`STATES.CONTENTVALID != y`). This statement unpublishes content in States where the value of Publishable is N, which is correct, but it will also unpublish content in States where the value of Publishable is I, which is not correct behavior.

To correct this problem, any unpublish Content List resource that uses the statement `STATES.CONTENT != y` must be updated with the statement `STATES.CONTENTVALID = n`.

Web Server Security Provider and Single Sign-on

The Rhythmyx Server Administrator has been enhanced to allow you to maintain Web Server Security Providers. Web server security providers on J2EE Web servers can now support Single Sign-on mode. This mode allows users to use a single log in to access Rhythmyx through a Web server, rather than having to log in twice (once to the Web server and then again to Rhythmyx). Single Sign-on mode requires some minor changes to the Rhythmyx servlet on the J2EE Web server.

For details about the new Web server security provider, see “Web Server Security Provider” in the Rhythmyx Server Administrator online Help.

For details about Single Sign-on, see “Implementing Single Sign-on” in the Rhythmyx Server Administrator online Help.

For details about changes to the Rhythmyx servlet to support Single Sign-on, see “Interfacing to Rhythmyx as a Servlet on J2EE Servers” in the Rhythmyx Server Administrator online Help.

Upgrading to Rhythmyx Version 5.5 Features

Rhythmyx Version 5.5 introduces a variety of new features. For details about all new Version 5.5 features, see “What’s New in Rhythmyx Version 5.5”. Implementer intervention is required to gain the full benefit of the following new features:

- WebDAV
- WebImageFX Image editor
- Text extraction
- Directory Services
- Accessibility enhancements

In addition, if you install the full-text search engine, you will need to create the search indices and you will also need to be aware of the interaction between the optional full-text search engine and existing saved searches.

For Rhythmyx Version 5.5, the recommended minimum version of the Java plugin has changed...

Finally, the Ektron eWebEditPro DHTML editor has been upgraded from Version 2.0 to Version 4.1

WebDAV

Rhythmyx Version 5.5 introduces support for WebDAV to link Rhythmyx Folders to working directories in your network. Using WebDAV, Content Contributors can automatically manage image- and file-type Content Items simply by working with them in their system directories. The system directories are linked to Rhythmyx Folders by WebDAV servlets.

WebDAV requires some manual implementation to link the system directories to the Rhythmyx Folders. For details, see *Implementing WebDAV in Rhythmyx*.

WebImageFX

Rhythmyx Version 5.5 incorporates the WebImageFX graphics editor from Ektron to provide basic image editing functionality. The WebImageFX editor, like the eWebEditPro DHTML text editor, is available as a Content Editor Control that can be added to Image Content Types. This control must be added to a Content Editor to make it available to Content Contributors. For details, see “sys_WebImageFX and the WebImageFX Editor” in the Content Editor Control appendix of *Implementing Content Editors*.

Text Extraction

The Convera RetreivalWare full-text search engine available with Rhythmyx 5.5 provides facilities Rhythmyx can access to extract text from binary files. A special Java extension, sys_TextExtraction, can be attached to a Content Editor resource to provide this functionality. For details about using the text extraction capabilities, see "Text Extraction" in *Implementing Content Editors*.

Directory Services

Directory services provides functionality that allows you to recover user attributes from a directory server, such as a LDAP server or a Microsoft ActiveDirectory server. You can use these attributes as part of user authentication, or use them as data for Rhythmyx processing.

To use Directory Services, you must define directory services configurations in the Rhythmyx Server Administrator. For details, see "Maintaining Directory Services Configurations" in the Rhythmyx Server Administrator Help.

Accessibility Enhancements

Rhythmyx Version 5.5 introduces a number of enhancements to improve accessibility for users with disabilities. Most of these enhancements require no additional implementation. An enhancement has been added to the Content Editor field properties, however, that allows you to add an accelerator key to each field in a Content Editor. For details, see "Field Maintenance Dialogs" in *Implementing Content Editors*.

Revised HTML Search Pages

New Search pages are available for the searches launched from Active Assembly, or when inserting inline links. These searches are rendered using HTML rather than using the search applet. The new search pages have been refactored to enhance their accessibility to users requiring assistive devices.

When you upgrade, the inline search page is automatically upgraded to use the new search page, but the Active Assembly search page is not. If you want to use the new pages, you must update the Active Assembly search component to use the correct Rhythmyx application. To update the Active Assembly search component:

5. In Content Explorer, choose the System tab.
6. In the Left Navigation, under the Components option, click the By Name link.
7. Click on the rcsearch link. (You may have to scroll through a few pages of component listings to reach this link.)
8. Change the value in the **URL** field to *../sys_searchSupport/getQuery.html*.
9. Click the [Save] button.
10. In the Workbench, restart the sys_rcSupport application.

If you want to return to the original Active Assembly search page, use the same procedure, but in step 4., change the value in the URL field to *../sys_relatedSearch/relatedsearch.html*.

Full-text Search Engine

The Convera RetrievalWare full-text search engine is an optional feature you can install to provide a robust, full-text alternative to the light-weight database search engine available with Rhythmyx. If you choose to install the full-text search engine, you will need to index your Content before searches will return results. Indexing can take several hours. Consider this fact when scheduling your upgrade installation.

To create search indices, enter the command:

```
search index type id
```

Where *id* is the Content Type ID of the Content Type you want to index. If you do not provide a value (in other words if you only enter the command `search index type`) all Content Types in your system will be indexed.

Any existing saved searched created using the light-weight database search engine will continue to exist until you delete them manually. Once these saved searches are deleted, you will not be able to recreate them.

The existing saved searches will continue to use the original, light-weight search functionality, but no full-text search functionality will be available with them. To take advantage of the full-text search functionality, you must create new saved searches after upgrading to the full-text search engine.

Minimum Recommended Version of the Java Plugin

The recommended minimum version of the Java plugin for Rhythmyx 5.5 is 1.4.2_04.

Upgrading the Ektron eWebEditPro Text Editor

NOTE: In Rhythmyx 5.7, a new rich text DHTML editor is introduced, and conversion to that Editor is recommended. For details, see "Ephox EditLive! for Java DHTML Editor" on page 49.

When upgrading from Version 4.5 of Rhythmyx to Version 5.7, the Ektron eWebEditPro inline text editor is upgraded from Version 2.0 to Version 4.1. The license for eWebEditPro is integrated into the sys_eWebEditPro control, but the license for Version 2.0 and Version 4.1 are incompatible.

Standard practice with the eWebEditPro control has been to overwrite only the sys_resources/ewebeditpro directory when upgrading Rhythmyx, leaving the rx_resources/ewebeditpro directory untouched to allow customers to upgrade the control at their convenience. A licensing change between Version 2.0 and 3.0 of eWebEditPro makes this impossible, however. Therefore, the Rhythmyx installer performs the following processing:

1. The sys_resources/ewebeditpro directory is overwritten during upgrade as usual.
2. The installer looks at the rx_resources/ewebeditpro directory for the file `sys_rxektron.properties`. If this file is missing or corrupt, or if the version of eWebEditPro specified in the file is different from the Version Rhythmyx is installing, the installer renames the directory

rx_resources/ewebeditpro/buildid_timestamp, and installs a new rx_resources/ewebeditpro directory with the new version of eWebEditPro.

The next time a user accesses a Content Item that uses the eWebEditPro control, the updated version of eWebEditPro will be downloaded to their system.

Upgrading to Version 5.7 Features

Rhythmyx Version 5.7 and earlier releases introduced a number of new features (white-box assembler, copying enhancements, and enhancements to Multi-server Manager), but these features do not require new implementation. (Most of them simplify or improve the process of implementation.). Two changes may require user intervention:

- The Tomcat Web application server is upgraded.
- A new rich text DHTML editor is introduced.

Tomcat Web Server Upgrade

In Rhythmyx 5.7, the Tomcat Web application server is upgraded to Version 5.5.7 In the process of this upgrade, the following files are overwritten:

- server.xml
- web.xml

Backups of these files are automatically created during the upgrade process. If you have made any modifications to these files, you will need to migrate the change from the backup file to the new version manually. The most common modifications to the server.xml are :

- add JNDI properties to the server.xml file in systems that use a JNDI directory server (and LDAP server or Microsoft Active Directory); and
- adding SSL port and keystore entries to systems using SSL.

Compare the new server.xml to the backup to ensure that you have not lost any important configuration settings.

Any additional web applications you may have implemented will be preserved, but should be tested to ensure that they still function as expected with Tomcat 5.5.7.

The cacerts file in the Tomcat JRE are also overwritten during the upgrade. This new cacerts file differs significantly from the earlier cacerts file, so you cannot rely on a backup of the file. You can, however, re-import the existing key from the Rhythmyx server into the new Tomcat JRE cacerts file. You will also need to export a key from the new Tomcat JRE and import that key into the Rhythmyx server JRE before you will be able to use SSL to publish.

If you are using database publishing via Tomcat you may also have made modifications, adding `<Resource>` elements to the server.xml. You must translate these elements into new `<Resource>` elements in `<Rxroot\AppServer\webapps\RxServices\META-INF\context.xml`. The resource parameters, defined in <Parameter> children must be translated as attributes of the `<Resource>` element.

If you use Microsoft SQL Server or Oracle as your target database, the new <Resource> element have already been defined for you. You only need to uncomment the appropriate <Resource> element, and copy the values of the Resource parameters currently defined in server.xml and paste them as the value of the attributes of the <Resource> element in context.xml.

If you use Sybase or DB2 as your target database, you will have to translate the elements manually.

Example translation for Sybase:

Original (Tomcat 4.1.27):

```
<Resource name="jdbc/dbpubtarget" auth="Container"
type="javax.sql.DataSource"/>
<ResourceParams name="jdbc/dbpubtarget">
  <parameter>
    <name>username</name>
    <value>sa</value>
  </parameter>
  <parameter>
    <name>password</name>
    <value>demo</value>
  </parameter>
  <parameter>
    <name>driverClassName</name>
    <value>com.sybase.jdbc2.jdbc.SybDriver</value>
  </parameter>
<!-- optional
  <parameter>
    <name>driver</name>
    <value>sybase</value>
  </parameter>
-->
  <parameter>
    <name>url</name>
    <value>jdbc:sybase:Tds:localhost:5000</value>
  </parameter>
</ResourceParams>
```

Translated for Tomcat 5.5.7:

```
<Resource name="jdbc/dbpubtarget" auth="Container"
type="javax.sql.DataSource"
driverClassName=”com.sybase.jdbc2.jdbc.SybDriver” driver=”sybase”
url=”jdbc:sybase:Tds:localhost:5000”/>
```

Example translation for DB2

Original (Tomcat 4.1.27):

```
<Resource name="jdbc/dbpubtarget" auth="Container"
type="javax.sql.DataSource"/>
<ResourceParams name="jdbc/dbpubtarget">
  <parameter>
    <name>username</name>
    <value>sa</value>
  </parameter>
```

```
    <parameter>
      <name>password</name>
      <value>demo</value>
    </parameter>
    <parameter>
      <name>driverClassName</name>
      <value>COM.ibm.db2.jdbc.app.DB2Driver</value>
    </parameter>
    <parameter>
      <name>url</name>
      <value>jdbc:db2:dbpubtarget</value>
    </parameter>
  </ResourceParams>
```

Translated for Tomcat 5.5.7:

```
 <Resource name="jdbc/dbpubtarget" auth="Container"
type="javax.sql.DataSource"
driverClassName="COM.ibm.db2.jdbc.app.DB2Driver" driver="sybase"
url="jdbc:db2:dbpubtarget"/>
```

You can add other attributes to this element. For a complete list of elements valid for this element, and their effect, see `http://jakarta.apache.org/commons/dbcp/configuration.html`.

Ephox EditLive! for Java DHTML Editor

In Rhythmyx Version 5.7, a new rich text DHTML editor is introduced, Ephox EditLive! for Java. This control supersedes the Ektron eWebEditPro rich text DHTML editor.

You do not have to replace eWebEditPro with EditLive! You can continue to use eWebEditPro if you prefer. Note, however, that Percussion Software will only support the eWebEditPro control until November 2006.

NOTE: You cannot use both the EditLive! and the eWebEditPro controls in the same Content Editor. The two controls are incompatible, and you must use one or the other consistently within any Content Editor. You can, however, use the EditLive! control in one Content Edit and the eWebEditPro control in a different Content Editor.

To change the rich text DHTML editor for a field from the eWebEditPro control to the EditLive! control:

1. In the Rhythmyx Workbench, open the Content Editor application whose editor you want to change and double-click on the Content Editor resource.

   Rhythmyx displays the Content Editor Properties dialog with the current configuration data for the Content Editor.
2. Select the Content Editor field whose control you want to change and click the [Edit] button.

   Rhythmyx displays the Field Properties dialog.
3. In the **Control** field, click the drop list and choose *sys_EditLive*.
4. Rhythmyx automatically sets common parameters to the same values used for the sys_eWebEditPro control that was used for the field.
5. If you want to use a customized configuration file, or modify other parameters:

   a) Click the browse button next to the Control field.

      Rhythmyx displays the Display Control Properties for <field> dialog.

   b) Enter the parameters and associated values you want to assign to the control.

   c) Click [**OK**] to save your edits.
6. On the Field Properties dialog, click [**OK**] to save the modifications you made to the field.
7. On the Content Editor Properties dialog click [**OK**] to save the modification you made to the Content Editor.

The changes will take effect the next time you start your application. To see your changes, stop and restart the application, log into Rhythmyx, and activate the editor.