
MarkLogic Server

Release Notes

Release 4.2
October, 2010

Last Revised: 4.2-5, July, 2011

Copyright

© Copyright 2002-2012 by MarkLogic Corporation. All rights reserved worldwide.

This Material is confidential and is protected under your license agreement.

Excel and PowerPoint are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries. This document is an independent publication of MarkLogic Corporation and is not affiliated with, nor has it been authorized, sponsored or otherwise approved by Microsoft Corporation.

Contains LinguistX, from Inxight Software, Inc. Copyright © 1996-2006. All rights reserved. www.inxight.com.

Antenna House OfficeHTML Copyright © 2000-2008 Antenna House, Inc. All rights reserved.

Argus Copyright ©1999-2008 Icenit Technology Ltd. All rights reserved.

Contains Rosette Linguistics Platform 6.0 from Basis Technology Corporation, Copyright © 2004-2008 Basis Technology Corporation. All rights reserved.

This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit (<http://www.openssl.org/>) Copyright © 1995-1998 Eric Young (eay@cryptsoft.com). All rights reserved. Copyright © 1998-2001 The OpenSSL Project. All rights reserved.

Contains software derived from the RSA Data Security, Inc. MD5 Message-Digest Algorithm. Copyright © 1991-1992, RSA Data Security, Inc. Created 1991. All rights reserved.

Contains ICU with the following copyright and permission notice:

Copyright © 1995-2010 International Business Machines Corporation and others. All rights reserved. Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, provided that the above copyright notice(s) and this permission notice appear in all copies of the Software and that both the above copyright notice(s) and this permission notice appear in supporting documentation.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR HOLDERS INCLUDED IN THIS NOTICE BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

Except as contained in this notice, the name of a copyright holder shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Software without prior written authorization of the copyright holder.

Table of Contents

Release Notes

Copyright	2
1.0 Introduction	6
2.0 Installation and Upgrade	7
2.1 Supported Platforms	8
2.2 Upgrade Support	9
3.0 New Features in Release 4.2	10
3.1 Replication	10
3.2 XSLT	10
3.3 Application Services Features	11
3.3.1 Information Studio to Collect and Transform Content	11
3.3.2 Application Builder Enhancements	11
3.3.3 Search, Info, and InfoDev APIs	12
3.4 Database and Administrative Features	12
3.5 Local-Disk Failover	13
3.5.1 Compartment Security	13
3.5.2 Database Rollback	13
3.5.3 Standalone Properties	13
3.5.4 Language/Encoding Detection	13
3.5.5 Bulk Load Tuning Parameters	14
3.5.6 Range Index on List Types (NMTOKEN)	14
3.5.7 Expanded Host Timing Information in xdmp:server-status	14
3.5.8 Password Policy Plugins	14
3.5.9 Merge Priority Database Option	14
3.5.10 Range Index Optimize Database Option	14
3.5.11 Mac OS X Supported for Development Use	15
3.6 Search and Analytics Features	15
3.6.1 Faster Calculation of Aggregates	15
3.6.2 Results Clustering	15
3.6.3 Proximity Boosting in Relevance Calculations	16
3.6.4 cts:distinctive-terms and cts:similar-query	16
3.6.5 Search Plan Information With xdmp:plan	16
3.6.6 Term Frequency Normalization	16
3.6.7 cts:document-fragment-query	17
3.6.8 Fast Sampling With score-random	17

3.6.9	Cardinality Selection with min-occurs and max-occurs	17
3.7	Welcome Page	17
3.8	Amazon ec2 Management API	17
3.9	XQuery Enhancements	17
3.9.1	xdmp:output Prolog Options For Serialization	18
3.9.2	New Built-In Functions	18
3.9.3	Import XSLT Stylesheets and Use Imported Functions and Variables ...	18
3.10	Documentation Enhancements	18
3.10.1	New Scripting Administrative Tasks Guide	19
3.10.2	Expanded Getting Started Guide	19
3.10.3	New Flexible Replication Guide	19
3.10.4	Expanded Application Services Documentation	19
3.10.5	Documentation for the New 4.2 APIs	19
3.10.6	XSLT and XQuery	19
4.0	Known Incompatibilities with Previous Releases	20
4.1	Search Incompatibilities	20
4.1.1	Scores of Searches With cts:properties-query Matches are Higher	21
4.1.2	cts:contains Queries with a cts:properties-query and a Range Query Now Require the Needed Indexes 21	
4.1.3	Default cts:near-query Distance Now 10	21
4.1.4	cts:train Uses Different Default Terms	21
4.2	Unicode Table Updated	21
4.3	XCC Jar Filed Renamed With Version Number	22
4.4	New mimetypes Should be Merged if You Have Customized Mimetypes	22
4.5	Default Serialization Changes for Indenting of Typed and Untyped XML (Changed in 4.2-1 to 4.2-3, the same as 4.1 in 4.2-4 and Later) 23	
4.5.1	4.1 Behavior	23
4.5.2	4.2-1 to 4.2-3 Behavior	24
4.5.3	4.2-4 and Later Behavior	25
4.6	Serialization Change For Default Attributes	26
4.7	Serialization Changes For XML Declaration	27
4.8	Signature Change for admin:database-geospatial-element-index and admin:data- base-geospatial-element-child-index 27	
4.9	Set Journaling on Shared-Disk Failover Databases to Strict	28
4.10	Linux Requirements Change	28
4.11	Upgrade and Reindex Required	28
4.12	Date Values With 5 or More Year Digits No Longer Allowed	28
4.13	Empty Function Body For XQuery Function Throws Syntax Error	29
4.14	XCC AutoCommit Methods Deprecated and Cause Compile Warnings	29
4.15	Incompatibilities Between 4.0 and 4.1	29
4.15.1	Forest Updates-Allowed Admin API Signature Changes	30
4.15.2	xdmp:eval/xdmp:invoke Changes for xdmp:set-response-code and Other App Server Built-In Functions 30	
4.15.3	Regular Expression Changes	30
4.15.4	Spelling Dictionaries are Now Required to be in the Spell Namespace ...	31

4.15.5	spell:suggest Now Returns Suggestions Even If Word Is In the Dictionary	
	31	
4.15.6	nobody User Has app-user Role	32
4.15.7	xdmp:directory-create Throws Exception if Directory Exists	32
4.15.8	XCC Now Requires Java 1.5 or Greater	32
4.15.9	Output Differences in xdmp:query-meters, xdmp:host-status, and xdmp:forest-status Beginning in 4.1-7	32
5.0	Other Notes	33
5.1	Memory and Disk Space Requirements	33
5.2	Compatibility with XQuery Specifications	34
5.3	XQuery Extensions	34
5.4	Documentation	34
5.5	Browser Requirements	36
5.6	Support	36

1.0 Introduction

MarkLogic Server 4.2 is a major release that includes many new features. The new features are described in “New Features in Release 4.2” on page 10. The following lists the major features with links to where they are described:

- [Replication](#)
- [XSLT](#)
- [Application Services Features](#)
 - [Information Studio to Collect and Transform Content](#)
 - [Application Builder Enhancements](#)
 - [Search, Info, and InfoDev APIs](#)
- [Database and Administrative Features](#)
 - [Compartment Security](#)
 - [Database Rollback](#)
 - [Standalone Properties](#)
 - [Language/Encoding Detection](#)
 - [Bulk Load Tuning Parameters](#)
 - [Range Index on List Types \(NMTOKEN\)](#)
- [Search and Analytics Features](#)
 - [Faster Calculation of Aggregates](#)
 - [Results Clustering](#)
 - [Proximity Boosting in Relevance Calculations](#)
 - [cts:distinctive-terms and cts:similar-query](#)
 - [Search Plan Information With xdmp:plan](#)
 - [Term Frequency Normalization](#)
 - [cts:document-fragment-query](#)
 - [Fast Sampling With score-random](#)
 - [Cardinality Selection with min-occurs and max-occurs](#)
- [Welcome Page](#)
- [Amazon ec2 Management API](#)
- [XQuery Enhancements](#)
- [Documentation Enhancements](#)

If you are upgrading from 4.1, some applications will require minor changes to run correctly on 4.2. For details, see “Known Incompatibilities with Previous Releases” on page 20.

For a list of bugs fixed in the latest maintenance release and a list of known bugs, see the MarkLogic Technical Support website at <http://support.marklogic.com> (supported customers only).

2.0 Installation and Upgrade

This chapter describes the supported platforms and upgrade paths for MarkLogic Server, and has the following sections:

- [Supported Platforms](#)
- [Upgrade Support](#)

2.1 Supported Platforms

This release of MarkLogic Server is supported on the following platforms:

- Microsoft Windows Server 2008 (x86), Microsoft Windows 2003 Server (x86), Microsoft Windows XP SP2, Microsoft Windows Vista 32-bit Edition (x86)*
- Microsoft Windows Server 2008 (x64), Windows 2003 Server 64-bit Edition (x64), Windows Vista 64-bit Edition (x64)*
- Sun Solaris 10 (64-bit SPARC)
- Sun Solaris 10 (x64)
- Red Hat Enterprise Linux 5 (x86)** ***
- Red Hat Enterprise Linux 5 (x64)** *** ****
- CentOS 5 (x64)** ***
- SUSE Linux Enterprise Server 11 (x64)** ***
- Mac OS X 10.6*****

* Microsoft Windows Vista is supported for development only.

If MarkLogic Server fails to start up on Windows with the error “the application failed to initialize properly (0xc0150002)”, then a dependency is missing from your environment and you need to download and install one of the following DLLs:

32-bit versions of Windows require the DLL at the following link:

<http://www.microsoft.com/downloads/details.aspx?familyid=200B2FD9-AE1A-4A14-984D-389C36F85647&displaylang=en>.

64-bit versions of Windows require the following DLL:

<http://www.microsoft.com/downloads/details.aspx?FamilyID=eb4ebe2d-33c0-4a47-9dd4-b9a6d7bd44da&DisplayLang=en>.

** The deadline I/O scheduler is required on Red Hat Linux platforms. The deadline scheduler is optimized to ensure efficient disk I/O for multi-threaded processes, and MarkLogic Server can have many simultaneous threads. For information on the deadline scheduler, see the Red Hat documentation (for example, <http://www.redhat.com/magazine/008jun05/features/schedulers/>).

***The `redhat-lsb`, `glibc`, and `gdb` packages are required on Red Hat Linux. Additionally, on 64-bit Red Hat Linux, both the 32-bit and the 64-bit `glibc` packages are required.

****Red Hat Linux 5 (x64) is also supported in a VMWare ESX 3.0.2, ESX 3.5.3, or ESX 4.0 (installed on bare metal) environment.

*****Mac OS X is supported for development only. Conversion (Office and PDF) and entity enrichment are not available on Mac OS X. Mac OS X 10.6 (Snow Leopard) on a 64-bit capable processor is required (<http://support.apple.com/kb/HT3696>).

2.2 Upgrade Support

This section describes upgrade support to 4.2. For details on installing MarkLogic Server and for the upgrade procedure, see the *Installation Guide*.

Upgrading is supported from 4.0-1 or later. If you are running a release prior to 4.0, you must first upgrade to 4.0 or 4.1 before upgrading to 4.2. If you are upgrading an Enterprise Edition cluster, you must first upgrade the node in which the Security database forest is located before you upgrade other nodes in the cluster.

An upgrade from 4.0 or 4.1 to 4.2 will reindex any databases that have `reindex enable` set to `true`. If you choose not to reindex your databases, they will run in either 3.0, 3.1, 3.2, 4.0, or 4.1 compatibility mode, depending on the version of MarkLogic Server in which they were last loaded or reindexed. Running in compatibility mode will disable certain 4.2 features and may treat all content in the database as English language content. For details on database compatibility, see the *Installation Guide*.

There are some known incompatibilities between 4.1 and 4.2. You might need to make some minor code changes to your 4.1 applications before they can run correctly in 4.2. For details on the incompatibilities, see “Known Incompatibilities with Previous Releases” on page 20. For instructions on upgrading to 4.2, including information about database compatibility between 4.2 and 4.1, see the *Installation Guide*.

3.0 New Features in Release 4.2

This chapter describes the new features in Release 4.2 of MarkLogic Server. The feature descriptions are divided into the following categories:

- [Replication](#)
- [XSLT](#)
- [Application Services Features](#)
- [Database and Administrative Features](#)
- [Search and Analytics Features](#)
- [Welcome Page](#)
- [Amazon ec2 Management API](#)
- [XQuery Enhancements](#)
- [Documentation Enhancements](#)

3.1 Replication

MarkLogic Server 4.2 includes support for replication. Replication allows asynchronous, document-level, filtered replication between two domains (usually two databases or directories). Replication is highly configurable, providing a wide array of functionality to replicate all or part of the content of one database into another. Replication is an important part of any enterprise-grade disaster recovery solution. Replication is also useful when scaling out, to push information close to your users. Replication makes it easy to replicate a database (or any part of a database) on one MarkLogic Server cluster to another database on either the same or on another MarkLogic Server cluster. You use Admin Interface and the Admin API to configure flexible replication. You can configure what domains to replicate and apply filters to limit documents that are replicated based on arbitrary XQuery code. You can configure replication to push changes from the master or, in secure environments, to pull changes from the replica. A replication target configuration may also include a scheduled task to handle retries and zero day replication. All of these features provides an extremely flexible set of tools to replicate whichever documents that you need. Because of this flexibility, it is known as *flexible replication*. To enable replication, a license key that includes flexible replication is required. For details on replication, including detailed information on how to set it up, see the *Flexible Replication Guide*.

3.2 XSLT

MarkLogic Server 4.2 includes XSLT 2.0, with XSLT 1.0 compatibility mode. You can invoke XSLT using the `xdmp:xslt-invoke` and `xdmp:xslt-eval` functions. XSLT is a popular language for creating applications that transform XML from one structure to another. In MarkLogic Server 4.2, you can combine XQuery and XSLT in your application, using whichever language that best fits the task you are performing. You can import XQuery library modules into XSLT stylesheets,

making the functions and variables from the XQuery library available in your XSTL stylesheet, and you can import stylesheets into XQuery modules. All of the MarkLogic Server built-in functions are available in both XQuery and XSLT. For more details on XSLT in MarkLogic Server, see the *XQuery and XSLT Reference Guide*.

3.3 Application Services Features

4.2 expands MarkLogic Application Services to add Information Studio and expands on Application Builder. This section highlights some of the new features in Application Services and includes the following parts:

- [Information Studio to Collect and Transform Content](#)
- [Application Builder Enhancements](#)
- [Search, Info, and InfoDev APIs](#)

3.3.1 Information Studio to Collect and Transform Content

4.2 introduces Information Studio, which is an application designed to make it easy to load content into MarkLogic Server and create simple or complicated transformations to perform during the loading process. Information Studio provides easy-to-use tools to:

- Load content from a directory from the filesystem.
- Load content from a remote machine using a drag-and-drop interface.
- Create flows to define a multi-step loading process.
- Customize a loading process (a content *collector*) and make it available in the user interface.
- Create and configure databases you need for your content.
- Add arbitrary XSLT to create simple or complex *transformers*.
- Track the progress of large multi-step loads, making it easy to see progress and errors.

For details, see the *Information Studio Developer's Guide*.

3.3.2 Application Builder Enhancements

Application Builder was introduced in 4.1, and it makes it easy to build search and analytics applications. In 4.2, Application Builder has been expanded to include the following:

- The ability to customize an application and have the customization survive re-deployment via the Application Builder.
- The addition of element and properties constraints as constraint types.
- Many usability improvements.

For details on Application Builder, see the *Application Builder Developer's Guide*.

3.3.3 Search, Info, and InfoDev APIs

In addition to the Search API (which was introduced in 4.1), Application Services now includes XQuery libraries to load content and make it easy to programmatically perform common tasks such as creating a database. The Information Studio and the Information Studio Developer APIs allow you to script information studio tasks as well as provide APIs that make it easy to write collectors to load content and transformers to transform content.

In addition to the new `info` and `info-dev` libraries, the `search` library includes the following enhancements:

- Element constraints, which allow you to constrain against content within the scope of an element.
- Properties constraints, which allow you to constrain against matches in properties documents.
- A proximity operator to perform near queries (`wolf near fox`).
- Range constraint operators to constrain queries to ranges of values.
- Geospatial constraints to build facets out of geospatial data.

For details, see the *MarkLogic XQuery and XSLT Function Reference*, the *Information Studio Developer's Guide*, and [Search API: Understanding and Using](#) in the *Search Developer's Guide*.

3.4 Database and Administrative Features

The following database and administrative features are included in MarkLogic Server 4.2:

- [Local-Disk Failover](#)
- [Compartment Security](#)
- [Database Rollback](#)
- [Standalone Properties](#)
- [Language/Encoding Detection](#)
- [Bulk Load Tuning Parameters](#)
- [Range Index on List Types \(NMTOKEN\)](#)
- [Expanded Host Timing Information in `xdmp:server-status`](#)
- [Password Policy Plugins](#)
- [Merge Priority Database Option](#)
- [Range Index Optimize Database Option](#)
- [Mac OS X Supported for Development Use](#)

3.5 Local-Disk Failover

MarkLogic Server 4.2 includes a second type of failover: *local-disk failover*. In MarkLogic Server 4.1, you can set up a forest on a shared clustered filesystem and, if the primary host for the forest becomes unavailable, another host will take control of the forest. The type of failover in 4.1 is referred to as *shared-disk failover*. 4.2 adds *local-disk failover*, which allows you to configure other forests on other hosts, and those forests will maintain replicas of the primary forest. In the event of the host for the primary forest becoming unavailable, MarkLogic Server will automatically transfer control to the replica forest (that is, it will fail the forest over to the replica forest, which runs on another host in the cluster). For details on both shared-disk failover and local-disk failover, including a discussion of the pros and cons of each failover type, see the *Scalability, Availability, and Failover Guide*.

3.5.1 Compartment Security

4.2 extends the MarkLogic Server security model with *compartment security*. A compartment is a name associated with a role, and that name is used as an additional check when accessing documents in a database or when using URI privileges. If you do not use compartments, then there is no change in the security model. If you do use compartment security, then you can require that a user passes multiple permission checks to access a document. Without compartment security, if a user satisfies at least one permission check, then the user can access the document. To enable compartment security, a license key that includes compartment security is required. For details, see [Compartment Security](#) in the *Understanding and Using Security Guide*.

3.5.2 Database Rollback

You can rollback one or more forests in a database to a point in time using the new `xdmp:forest-rollback` built-in function. Doing a rollback to a point-in-time is much faster than doing a database restore, and it is a good way to reset the system to a known state. For more details, see the *MarkLogic XQuery and XSLT Function Reference* and [Rolling Back a Forest to a Particular Timestamp](#) in *Application Developer's Guide*.

3.5.3 Standalone Properties

4.2 allows you to have a properties document without having a corresponding document at the same URI. Previously, you needed to create the document, either before or at the same time as creating the properties document. A standalone properties document is useful retaining metadata about a deleted document or for storing information (such as RDF tuples) in a compact form. For details, see [Standalone Properties Documents](#) in the *Application Developer's Guide*.

3.5.4 Language/Encoding Detection

4.2 includes the `xdmp:encoding-language-detect` function, which provides an XML report with information about the language and encoding of a document. Additionally, `xdmp:document-load`, `xdmp:document-get`, `fn:unparsed-text`, and the `xdmp:http-*` functions all have an `<encoding>auto</encoding>` option which will automatically set the encoding during loading. For details, see [Specifying Encoding and Language During Loads](#) in the *Application Developer's Guide*.

3.5.5 Bulk Load Tuning Parameters

You can now configure transaction journal writes and transaction locks to be more lax for faster bulk loading. This is useful when you do not need transactional integrity because you can start over if something goes wrong. For details, see [Understanding the Locking and Journaling Database Settings for Bulk Loads](#) in the *Application Developer's Guide*.

3.5.6 Range Index on List Types (NMTOKEN)

You can now create range indexes on list types. You choose the type of the range index when creating the index, and each item in the list will be indexed as a single value.

3.5.7 Expanded Host Timing Information in `xmdp:server-status`

The `xmdp:server-status` built-in now includes more information about activity on each host in a cluster. It provides output to show the time the App Server spends communicating with each host in the cluster. The `xmdp:query-meters` and `xmdp:host-status` built-ins provide similar host-communication timings. For details on the output of `xmdp:server-status`, see *MarkLogic XQuery and XSLT Function Reference*.

3.5.8 Password Policy Plugins

MarkLogic Server 4.2 includes a plugin framework to allow you to register capabilities, and then implement those capabilities as XQuery modules which are evaluated by the plugin framework. You can use this framework to create plugins that check a password when it is set by a user. For example, you can write a plugin to check that the password has a certain number of characters or includes special characters, and if it does not, return an error. Some sample password plugin code is included in the `samples` directory. For details on the password plugin samples and on the plugin framework, see [Plugin Framework](#) in the *Application Developer's Guide*.

3.5.9 Merge Priority Database Option

There is new database option, `merge priority`, that allows you to set the CPU priority given to merges. To make merges get a lower CPU priority, set the option to `lower`, which will give merges a lower CPU priority than requests against that server. This can be a useful tool if database merges are slowing down queries on your system(s) too much. You can set the `merge priority` in the Admin Interface on the Merge Policy page (under the database page) or using the `admin:database-set-merge-priority` and `admin:database-get-merge-priority` Admin API functions.

3.5.10 Range Index Optimize Database Option

There is a new database option, `range index optimize`, which specifies whether range indexes should be optimized to make lexicon (and therefore facet) resolution as fast as possible (`facet-time`) or to use less memory (`memory-size`). This allows you to make the choice in the trade-off between speed and memory usage. If you do not need to conserve memory, then continue to use the default of `facet-time`.

3.5.11 Mac OS X Supported for Development Use

MarkLogic Server 4.2 is available for development use only on Mac OS X 10.6. It uses the native 64-bit capabilities of Mac OS X. Conversion and entity extraction are not available on Mac OS X. For details about requirements and installing MarkLogic Server on Mac OS X or on any of the supported platforms, see the *Installation Guide*.

3.6 Search and Analytics Features

The following are search-related features in MarkLogic Server 4.2:

- [Faster Calculation of Aggregates](#)
- [Results Clustering](#)
- [Proximity Boosting in Relevance Calculations](#)
- [cts:distinctive-terms and cts:similar-query](#)
- [Search Plan Information With xdmp:plan](#)
- [Term Frequency Normalization](#)
- [cts:document-fragment-query](#)
- [Fast Sampling With score-random](#)
- [Cardinality Selection with min-occurs and max-occurs](#)

3.6.1 Faster Calculation of Aggregates

The W3C XQuery standard aggregate functions (`fn:avg`, `fn:max`, `fn:min`, and `fn:sum`) are faster in 4.2 than in previous releases, and they are particularly faster when computing aggregates based on sequences returned from the lexicon function (for example, `cts:element-values`).

Additionally, the following new functions take sequences of values from the lexicon functions and take into account frequency (the number of occurrences per fragment) when calculating the aggregates:

- `cts:sum`
- `cts:avg`

For details on these functions, see the *MarkLogic XQuery and XSLT Function Reference*.

3.6.2 Results Clustering

4.2 introduces `cts:cluster`, which provides output that allows you to cluster search results into groups of similar documents with labels. You can use the output from `cts:cluster` to build a user interface that shows search results in categories, where results in a category are statistically similar to each other. For details, see [Results Clustering Using cts:cluster](#) in the *Search Developer's Guide*.

3.6.3 Proximity Boosting in Relevance Calculations

With MarkLogic 4.2 a search may optionally boost the score of a search result based on the proximity of terms; that is, terms found close to each other will score higher than terms found farther apart. There is a `distance-weight` option on the leaf-level `cts:query` constructors which allows you to specify the level of proximity boost for a query. For proximity boosting to work you must have the index option `word positions` set to `true`.

With MarkLogic 4.2 a search may optionally boost the score of a search result based on the proximity of terms; that is, terms found close to each other will score higher than terms found farther apart. There is a `distance-weight` option on the leaf-level `cts:query` constructors which allows you to specify the level of proximity boost for a query. To use proximity boosting, you must have the word positions For proximity boosting to work you must have the index option `word positions` database option set to `true`. For details, see [Proximity Boosting With the distance-weight Option](#) in the *Search Developer's Guide*.

3.6.4 `cts:distinctive-terms` and `cts:similar-query`

4.2 includes introduces `cts:distinctive-terms`, which allows you to find the most “relevant” terms that make the nodes similar. Additionally, you can now pass options to `cts:similar-query` to use the options introduced in `cts:distinctive-terms`. The `cts:distinctive-terms` options allow you to pass in database indexing options, which can affect the results by considering search terms applicable to those index options. For more information, see the *MarkLogic XQuery and XSLT Function Reference* and [Distinctive Terms and `cts:similar-query`](#) in the *Search Developer's Guide*.

3.6.5 Search Plan Information With `xdmp:plan`

4.2 includes the `xdmp:plan` function, which returns information on how a search or an XPath expression is evaluated. It returns information similar to the information logged by `xdmp:query-trace`, as well as information on the query terms that are selected. For more details, see the *Query Performance and Tuning Guide* and the *MarkLogic XQuery and XSLT Function Reference*.

3.6.6 Term Frequency Normalization

When documents are very different in size, larger documents tend to contain more occurrences of any term than shorter ones, which causes large documents to float to the top of a relevance-ranked search results list. To balance this effect `tf` (term-frequency) may be normalized according to the length of the document. This feature lets you control how much `tf` is affected by result length, to tune your search results ordering. The `tf normalization` option on the database configuration controls this setting.

In 4.1, an option to set the term frequency normalization (`tf normalization`) to `unscaled-log` was introduced; `unscaled-log` does no scaling based on document size. In 4.2, there are four new intermediate settings, which have varying degrees of scaling. If you change this setting in the database and `reindexer enable` is set to `true`, then the database will reindex. Setting the `tf normalization` option to any of the scaled options will take into account the size of the document, not just the term frequency in the document, as part of the relevance calculation. The default value is `scaled-log`, which has the strongest amount of scaling for document size.

3.6.7 `cts:document-fragment-query`

4.2 introduces a new `cts:query` constructor, `cts:document-fragment-query`. Just as `cts:properties-query` allows you to add property constraints when searching against documents, `cts:document-fragment-query` allows you to add document constraints when searching against properties.

3.6.8 Fast Sampling With `score-random`

A new scoring algorithm for `cts:search`, `score-random`, allows you to sort a search in random order, making it very efficient to get random samples from a search result.

3.6.9 Cardinality Selection with `min-occurs` and `max-occurs`

You can now constrain searches to return results only if the search terms appear a minimum or a maximum number of times. There are new options to the `cts:query` constructors to support this functionality.

3.7 Welcome Page

4.2 includes a new Welcome Page running on the Docs App Server on port 8000. The Welcome Page includes links to Application Services, links to the documentation, and other useful information. It is a useful starting point for using MarkLogic Server. The XQuery use-cases demo is still available under the Welcome Page. For more details, see [MarkLogic Server Entry Page](#) in *Getting Started with MarkLogic Server*.

3.8 Amazon ec2 Management API

4.2 includes an XQuery API used to manage Amazon ec2 hosts. The API allows you to write XQuery scripts to configure instances on ec2. For details on the API, see the Module: Elastic Compute Cloud API documentation in the *MarkLogic XQuery and XSLT Function Reference*.

3.9 XQuery Enhancements

This section details various XQuery enhancements in 4.2, and includes the following enhancements:

- [xdmp:output Prolog Options For Serialization](#)
- [New Built-In Functions](#)
- [Import XSLT Stylesheets and Use Imported Functions and Variables](#)

3.9.1 **xdmp:output Prolog Options For Serialization**

You can set an option in the XQuery prolog to set the serialization mode for a query. These options support the various XSLT serialization methods from within XQuery. The serialization options include indenting, HTML, and many other options. For details, see [Declaring Options](#) in the *XQuery and XSLT Reference Guide*.

3.9.2 **New Built-In Functions**

There are many new built-in functions in 4.2, available in both XQuery and XSLT, including the following:

- `xdmp:parse-dateTime`
- `xdmp:parse-yymmdd`
- `xdmp:lock-for-update`
- `xdmp:get-server-field`, `xdmp:set-server-field`, `xdmp:get-server-field-names`
- and many more...

For details, see the *MarkLogic XQuery and XSLT Function Reference*.

3.9.3 **Import XSLT Stylesheets and Use Imported Functions and Variables**

4.2 allows you to import an XSLT stylesheet into your XQuery program. You then can use any functions and variables defined in the stylesheet from within the XQuery program. Similarly, you can import an XQuery library into an XSLT stylesheet and use its functions and variables in the stylesheet. For details, see [MarkLogic Server Enhanced XQuery Language](#) in the *XQuery and XSLT Reference Guide*.

3.10 **Documentation Enhancements**

MarkLogic Server includes a comprehensive set of documentation. 4.2 includes the following enhancements to the documentation:

- [New Scripting Administrative Tasks Guide](#)
- [Expanded Getting Started Guide](#)
- [New Flexible Replication Guide](#)
- [Expanded Application Services Documentation](#)
- [Documentation for the New 4.2 APIs](#)
- [XSLT and XQuery](#)

For a list of what is in the documentation, see “Documentation” on page 34. To download the documentation, go to developer.marklogic.com/pubs.

3.10.1 New Scripting Administrative Tasks Guide

The information about using the Admin API and other APIs to script administrative tasks has been expanded and moved from the *Administrator's Guide* to the new *Scripting Administrative Tasks Guide*. This guide describes how to use the various administrative APIs and includes sample code for a wide variety of activities, including scripting CPF configurations.

3.10.2 Expanded Getting Started Guide

The *Getting Started with MarkLogic Server* guide has been expanded to include more exercises appropriate to a new user of MarkLogic Server.

3.10.3 New Flexible Replication Guide

A new guide, the *Flexible Replication Guide*, describes in detail how to setup and use replication.

3.10.4 Expanded Application Services Documentation

The Application Services documentation has been expanded as follows:

- Additions to the *Application Builder Developer's Guide*
- A new *Information Studio Developer's Guide*
- New API documentation for the Information Studio and Information Studio developer APIs.

3.10.5 Documentation for the New 4.2 APIs

There are several hundred new XQuery/XSLT functions available in 4.2. For details, see the *MarkLogic XQuery and XSLT Function Reference*.

3.10.6 XSLT and XQuery

The *XQuery Reference Guide* has been renamed to *XQuery and XSLT Reference Guide*, and describes both the MarkLogic extensions to XQuery as well as the extensions to XSLT. Additionally, this guide has background information about XQuery, XPath, and XSLT.

4.0 Known Incompatibilities with Previous Releases

The vast majority of applications implemented on MarkLogic Server 4.1-* will run either without modifications or with very minor modifications in Release 4.2. There are, however, a number of changes that will cause compatibility issues with 4.1 applications. This section describes those incompatibilities and includes the following topics:

- [Search Incompatibilities](#)
- [Unicode Table Updated](#)
- [XCC Jar Filed Renamed With Version Number](#)
- [New mimetypes Should be Merged if You Have Customized Mimetypes](#)
- [Default Serialization Changes for Indenting of Typed and Untyped XML \(Changed in 4.2-1 to 4.2-3, the same as 4.1 in 4.2-4 and Later\)](#)
- [Serialization Change For Default Attributes](#)
- [Serialization Changes For XML Declaration](#)
- [Signature Change for admin:database-geospatial-element-index and admin:database-geospatial-element-child-index](#)
- [Set Journaling on Shared-Disk Failover Databases to Strict](#)
- [Linux Requirements Change](#)
- [Upgrade and Reindex Required](#)
- [Date Values With 5 or More Year Digits No Longer Allowed](#)
- [Empty Function Body For XQuery Function Throws Syntax Error](#)
- [XCC AutoCommit Methods Deprecated and Cause Compile Warnings](#)
- [Incompatibilities Between 4.0 and 4.1](#)

4.1 Search Incompatibilities

Most search code will function identically in 4.2 as it did in 4.1, but there are a few cases where the behavior might be slightly different. This section describes the incompatibilities in search features between 4.1 and 4.2, and includes the following incompatibilities:

- [Scores of Searches With cts:properties-query Matches are Higher](#)
- [cts:contains Queries with a cts:properties-query and a Range Query Now Require the Needed Indexes](#)
- [Default cts:near-query Distance Now 10](#)
- [cts:train Uses Different Default Terms](#)

4.1.1 Scores of Searches With `cts:properties-query` Matches are Higher

Searches that match a `cts:properties-query` in a `cts:and-query` will have higher scores than in 4.1. In 4.2, the score contributions of the `cts:properties-query` and the other `cts:query` constructors that operate on the document are added together, whereas in 4.1 the score was the maximum of the property and document scores.

4.1.2 `cts:contains` Queries with a `cts:properties-query` and a Range Query Now Require the Needed Indexes

If you have an expression that uses `cts:contains` with a `cts:properties-query` (or a `cts:document-fragment-query`), those expressions will now use the indexes to perform index resolution. Therefore, any `cts:query` constructors used with `cts:properties-query` or `cts:document-fragment-query` will throw an exception if any needed indexes are not available. For example, the following query will now throw an exception if there is no range index on the element `my-element`:

```
cts:contains(//a, cts:properties-query(  
  cts:element-range-query(xs:QName("my-element"), "=", "my-value")))
```

In 4.1, this query would complete even if there was no range index, but it would have to gather all of the needed fragments in order to perform the `cts:properties-query`.

4.1.3 Default `cts:near-query` Distance Now 10

The default distance for `cts:near-query` is now 10. Previously, the default was 100. If your applications use `cts:near-query` without specifying a `$distance` parameter, they will only match words within a 10 word distance by default. If you want them to match according to the old 100 word value, you must add a value of 100 to the `$distance` parameter.

4.1.4 `cts:train` Uses Different Default Terms

The default database terms that `cts:train` uses have changed in 4.2. In 4.2, the defaults are `db:stemmed-searches`, `db:fast-phrase-searches`, `db:fast-element-word-searches`, and `db:fast-element-phrase-searches`. In 4.1, the default was only `db:word-searches`. If you want the same behavior as 4.1, you can modify the options to `cts:train` to use `db:word-searches`. For most data, however, the new defaults should give better classifications.

4.2 Unicode Table Updated

The Unicode table has been updated to the following Unicode character mappings:

<http://unicode.org/versions/Unicode5.2.0/>

These are minor changes, but they might cause some minor incompatibilities in applications where a character was mapped to one set before and is mapped to another set now. For example, it can cause some regular expressions that matched before to not match now, or vice-versa.

4.3 XCC Jar File Renamed With Version Number

The XCC jar files, used for Java client programs to connect to MarkLogic Server, now include the version number as part of the file name. The names are of the form `marklogic-xcc-4.2.1.jar`. Previously, the jar files were all named `xcc.jar`, regardless of the version. When you use the new jar file in your existing java programs, you will have to either rename the jar file before copying it to your application or modify the Java startup scripts to use the new name.

4.4 New mimetypes Should be Merged if You Have Customized Mimetypes

MarkLogic Server 4.2 includes the following new mimetype values:

Name	Extensions	Format	Description
<code>application/atom+xml</code>	N/A	<code>xml</code>	Mimetype for Atom Feeds.
<code>text/cache-manifest</code>	<code>manifest</code>	<code>text</code>	Mimetype for cached manifest files for offline web applications.
<code>text/csv</code>	<code>csv</code>	<code>text</code>	Mimetype for comma separated value files.
<code>text/html-sandboxed</code>	<code>sandboxed</code>	<code>text</code>	Mimetype for sandboxed html files.
<code>text/ping</code>	N/A	<code>text</code>	Mimetype for ping feeds.

If you upgraded to 4.2 from a previous version of MarkLogic Server and if you have ever customized your mimetypes (for example, using the MIME Types Configuration page of the Admin Interface), the upgrade will not automatically add the new mimetypes to your configuration. If you have not added any mimetypes, then the new mimetypes will be automatically added during the upgrade. You can check if you have these mimetypes configured by going to the Mimetype page of the Admin Interface and checking if the above mimetypes exist. If they do, then there is nothing you need to do.

If you do not have the above mimetypes after upgrading to 4.2, you can manually add the mimetypes to your configuration using the Admin Interface. To manually add the configuration, perform the following:

1. Open the Admin Interface in a browser (for example, open <http://localhost:8001>).
2. Navigate to the Mimetypes page, near the bottom of the tree menu.
3. Click the Create tab.
4. Enter the name, the extension, and the format for the mimetype (see the table above).
5. Click OK.
6. Repeat the preceding steps for each mimetype in the above table.

4.5 Default Serialization Changes for Indenting of Typed and Untyped XML (Changed in 4.2-1 to 4.2-3, the same as 4.1 in 4.2-4 and Later)

The default serialization of untyped XML (that is, XML for which there is no in-scope schema) has changed in 4.2. Additionally, the behavior changed again in 4.2-4. This section describes these changes as follows:

- [4.1 Behavior](#)
- [4.2-1 to 4.2-3 Behavior](#)
- [4.2-4 and Later Behavior](#)

The examples below show how you can use the new serialization capabilities in 4.2 to control the XML serialization. Note that you can add any of the 4.2 `xm:output` prolog statements to your 4.1 code, and they will simply be ignored in 4.1 (because `declare option` is designed to ignore unknown options, and the `xm:output` option does not exist in 4.1, but does in 4.2).

Note: If you are using CQ, the `xm:output` options shown in the examples below will be ignored and instead use the defaults (because it performs an `xm:eval` of your query); to see the results of the `xm:output` options, evaluate the query directly against an App Server.

4.5.1 4.1 Behavior

In 4.1, untyped XML is not indented when serialized.

```
(: in 4.1 :)
xquery version "1.0-ml" ;

<test>
  <indented>hello</indented><not/></test>

=>
<test>
  <indented>hello</indented><not/></test>
```

4.5.2 4.2-1 to 4.2-3 Behavior

In 4.2-3 and earlier 4.2 releases, untyped XML is indented by default. Additionally, 4.2 included the `xdmp:output indent` option which allows you to control the indentation of XML. In 4.2-3 and earlier, the `indent` option controls both typed and untyped XML (XML with a schema and without).

```
(: in 4.2-1 to 4.2-3 :)
xquery version "1.0-ml" ;

<test>
  <indented>hello</indented><not/></test>

=>
<test>
  <indented>hello</indented>
  <not/>
</test>

(: in 4.2-1 to 4.2-3 (emulates the 4.1 behavior) :)
xquery version "1.0-ml" ;
declare option xdmp:output "indent=no" ;

<test>
  <indented>hello</indented><not/></test>

=>
<test>
  <indented>hello</indented><not/></test>

(: in 4.2-1 to 4.2-3 :)
xquery version "1.0-ml" ;
declare option xdmp:output "indent=yes" ;

<test>
  <indented>hello</indented><not/></test>

=>
<test>
  <indented>hello</indented>
  <not/>
</test>
```

4.5.3 4.2-4 and Later Behavior

In 4.2-4 and later releases, the `indent` option no longer applies to untyped XML (XML with no in-scope schema); it only applies to typed XML (XML with an in-scope schema). An additional `xdmp:output` option—"indent-untyped=yes" or "indent-untyped=no"—controls the serialization of untyped XML. Additionally, you can enable trace events and add the trace event `OUTPUT INDENT UNTYPED` to the diagnostics page in the Admin Interface to make the default for untyped XML serialiazation be indented (like it was in 4.2-1 to 4.2-3).

```
(: in 4.2-4 and later :)
xquery version "1.0-ml" ;

<test>
  <indented>hello</indented><not/></test>

=>
<test>
  <indented>hello</indented><not/></test>

(: in 4.2-4 and later :)
xquery version "1.0-ml" ;
declare option xdmp:output "indent-untyped=no" ;

<test>
  <indented>hello</indented><not/></test>
=>
<test>
  <indented>hello</indented><not/></test>

(: in 4.2-4 and later :)
xquery version "1.0-ml" ;
declare option xdmp:output "indent-untyped=yes" ;

<test>
  <indented>hello</indented><not/></test>
=>
<test>
  <indented>hello</indented>
  <not/>
</test>

(: in 4.2-4 and later, OUTPUT INDENT UNTYED trace event enabled :)
xquery version "1.0-ml" ;

<test>
  <indented>hello</indented><not/></test>
=>
<test>
  <indented>hello</indented>
  <not/>
</test>
```

4.6 Serialization Change For Default Attributes

In 4.1, if an attribute is defined in a schema to have a default value, then documents that do not have the attribute are serialized with the default attribute added. In 4.2, the default attribute is not added to a document stored in the database when it is serialized (for example, with `xdrm:save`). It is, however, available in the DOM, and therefore if you create a copy of the document before serializing it, then the default attribute will be part of the serialized output. If you have applications that rely on the 4.1 behavior, you can make code changes that copy the document to a constructed node before serializing it. For example, if you have code in 4.1 that does the following (and relies on the default serialization of an attribute with a default value defined in a schema):

```
xdrm:save("/space/myfile.xml", fn:doc("/myfile.xml"))
```

Then in 4.2 you can change that code as follows, which results in the attribute serialized with the default value:

```
xdrm:save("/space/myfile.xml", document{ fn:doc("/myfile.xml") })
```

Additionally, beginning in 4.2-5, there is an additional option to the `xdrm:output` prolog option:

```
declare option xdrm:output "default-attributes = yes";
```

When set to `yes`, any default attributes are serialized. The default is `no`.

Note: If you are using CQ, the `xdrm:output` options shown above will be ignored and instead use the defaults (because it performs an `xdrm:eval` of your query); to see the results of the `xdrm:output` options, evaluate the query directly against an App Server.

Additionally, you can enable trace events and add the trace event `OUTPUT_DEFAULT_ATTRIBUTES` to the diagnostics page in the Admin Interface to make the default for default attribute serialization serialize attributes if the schema has a default value and the XML has no value (like it was in 4.1).

4.7 Serialization Changes For XML Declaration

In 4.1, when you serialize an XML document, MarkLogic does not serialize the XML declaration with the document. In 4.2, in conformance with the XML specification, MarkLogic includes, by default, the XML declaration when serializing an XML document. For example, in 4.2, the following query returns the XML declaration with the returned document:

```
xquery version "1.0-ml";

xdmp:set-response-content-type("text/plain"),
document { <foo/> }
=>
<?xml version="1.0" encoding="ASCII"?>
<foo/>
```

If you want the 4.1 behavior in 4.2, you can use the `xdmp:output "omit-xml-declaration=yes"` declaration in your XQuery prolog as in the following example:

```
xquery version "1.0-ml";
declare option xdmp:output "omit-xml-declaration = yes";

xdmp:set-response-content-type("text/plain"),
document { <foo/> }
=>
<foo/>
```

Note: If you are using CQ, the `xdmp:output` options shown above will be ignored and instead use the defaults (because it performs an `xdmp:eval` of your query); to see the results of the `xdmp:output` options, evaluate the query directly against an App Server.

Additionally, you can enable trace events and add the trace event `OUTPUT OMIT XML DECLARATION` to the diagnostics page in the Admin Interface to make the default for the XML declaration serialization not serialize the XML declaration (like it was in 4.1).

4.8 Signature Change for `admin:database-geospatial-element-index` and `admin:database-geospatial-element-child-index`

The signature has changed for the following functions in the Admin API:

- `admin:database-geospatial-element-index`
- `admin:database-geospatial-element-child-index`

The order of the last 2 arguments, and the `$point-format` argument is now last and is optional. If you have code that uses either of these functions, then see the function signatures in *MarkLogic XQuery and XSLT Function Reference* and determine if you need to change your code. If you use the `$point-format` argument, you will need to change the order of the last 2 arguments.

4.9 Set Journaling on Shared-Disk Failover Databases to Strict

In 4.2, there are two types of failover: shared-disk failover and local-disk failover. Shared-disk failover is the same as the failover available in 4.1. In 4.1, failover forests always use the `journaling` database option of `strict`, but you cannot manually set the journaling option in 4.1. In 4.2, MarkLogic Server allows you to manually set the `journaling` option at the database level, but the default is `fast`. If you are upgrading from 4.1 to 4.2 and are using failover (shared-disk failover, which was available in 4.1), to have the same behavior as 4.1, you should change the database option for `journaling` on any databases in which you have failover forests configured to `strict`. If you leave the option at the default of `fast`, your system will be protected against the MarkLogic Server process going down during transactions, but not protect against the computer going down.

For details on the journaling settings, see [Understanding the Locking and Journaling Database Settings for Bulk Loads](#) in the *Application Developer's Guide*. For details on failover, see *Scalability, Availability, and Failover Guide*.

4.10 Linux Requirements Change

To run 4.2 on Linux distributions, Red Hat 5 is required. MarkLogic Server will no longer start up on Red Hat 4. If you are using Red Hat 4, you must upgrade your operating system before upgrading MarkLogic Server. For details on system requirements, see “Supported Platforms” on page 8.

4.11 Upgrade and Reindex Required

When you log into the Admin Interface after installing 4.2, you will be prompted to upgrade the security database and the configuration files. The 4.2 security database and configuration files are not backwards compatible with 4.1. Make sure to do a full backup of your databases and data directory before upgrading. If you do not want to reindex after upgrading, turn off reindexing for each database in 4.1 before installing 4.2. For details and index compatibility, see [Upgrades and Database Compatibility](#) in the *Installation Guide*.

4.12 Date Values With 5 or More Year Digits No Longer Allowed

Beginning with 4.2-2, `xs:date` and `xs:dateTime` values can no longer have more than 4 digits for the year. Previously, MarkLogic sometimes allowed dates with more than 4 digits. The W3C XML schema specification does not require dates with years greater than 9999, and MarkLogic Server does not support such values. In 4.2-1 and earlier, however, MarkLogic Server did not consistently check for large year values.

If you have data with 5 or more digit year values in `xs:date` and `xs:dateTime` elements or attributes, you should change the data to conform 4-digit years. If you try to load content with 5 or more year digits and if there are range indexes on those elements or attributes, the load will fail beginning with 4.2-2. Similarly, if you reindex content with 5 or more year digits in `xs:date` and `xs:dateTime` values and there are range indexes on those values, documents that have such data will fail to reindex and an error will be logged to the `ErrorLog.txt` file.

4.13 Empty Function Body For XQuery Function Throws Syntax Error

Beginning with 4.2-2, an empty body in a function declaration is no longer allowed. The XQuery specification has always disallowed an empty function body, but prior to 4.2-2, MarkLogic Server allowed it (it implicitly made the function body an empty sequence).

If you have an empty function body, you will need to modify your code to explicitly put an empty sequence in the body.

For example, the following throws a syntax error in 4.2-2 and later:

```
xquery version "1.0-ml";  
  
declare function local:test () {};  
  
local:test ()
```

To correct the syntax error, add an empty sequence in the function body:

```
xquery version "1.0-ml";  
  
declare function local:test () { () };  
  
local:test ()
```

The correct syntax will also work in previous releases of MarkLogic Server.

4.14 XCC AutoCommit Methods Deprecated and Cause Compile Warnings

The XCC `Session.setAutoCommit` and `Session.getAutoCommit` methods are now deprecated, and beginning with 4.2-4, they will cause compilation warnings if used in XCC code. These methods did not provide any functionality, and future releases will not include these methods. If you have any code that uses these methods, you should remove them from your code. You do not have to remove the methods immediately, however, as they will continue to exist in 4.1 and 4.2 versions of XCC. However, these methods may not exist in future versions of XCC.

4.15 Incompatibilities Between 4.0 and 4.1

MarkLogic Server 4.2 allows you to upgrade either from 4.0 or 4.1. If you are upgrading from 4.0, there are some known incompatibilities between 4.0 and 4.1 that are documented in the 4.1 Release Notes. If you are upgrading from 4.1, you can skip this section. For convenience, the incompatibilities between 4.0 and 4.1 are repeated here, and are as follows:

- [Forest Updates-Allowed Admin API Signature Changes](#)
- [xdmp:eval/xdmp:invoke Changes for xdmp:set-response-code and Other App Server Built-In Functions](#)
- [Regular Expression Changes](#)
- [Spelling Dictionaries are Now Required to be in the Spell Namespace](#)

- [spell:suggest Now Returns Suggestions Even If Word Is In the Dictionary](#)
- [nobody User Has app-user Role](#)
- [xdmp:directory-create Throws Exception if Directory Exists](#)
- [XCC Now Requires Java 1.5 or Greater](#)
- [Output Differences in xdmp:query-meters, xdmp:host-status, and xdmp:forest-status Beginning in 4.1-7](#)

4.15.1 Forest Updates-Allowed Admin API Signature Changes

The following functions for the Admin API have different signatures from 4.0, and are therefore incompatible:

- `admin:forest-get-updates-allowed`
- `admin:forest-set-updates-allowed`

In 4.1, there are four values for `updates-allowed`; in 4.0, it was a boolean value. If you have code that uses these APIs, you must modify the code to use the new values. The `$value` parameter of `admin:forest-set-updates-allowed` can now have the following values:

- `all` (was `true` in 4.0)
- `delete-only` (was `false` in 4.0)
- `read-only`
- `flash-backup`

The new values reflect the ability to mark a forest read-only in 4.1.

4.15.2 xdmp:eval/xdmp:invoke Changes for xdmp:set-response-code and Other App Server Built-In Functions

Previously, `xdmp:eval` and `xdmp:invoke` ignored any calls to the various App Server Built-Ins that change or get state from the response or the request body (for example, `xdmp:set-response-code`, `xdmp:redirect-response`, and so on). In 4.1, these calls work in an `xdmp:eval` or an `xdmp:invoke` the same way they do when you run a query in any other context. If you have applications that relied on the old behavior (that is, applications that used App Server Built-In functions in an `xdmp:eval/xdmp:invoke` but expected them to be no-ops), then you might need to rewrite those `xdmp:eval` or `xdmp:invoke` statements.

4.15.3 Regular Expression Changes

The regular expression evaluation in 4.1 has been improved and is more efficient than in 4.0. It is also more conformant to the XQuery specification than 4.0. Some of these conformance changes will cause some regular expressions to behave differently in 4.1 than they did in 4.0. Regular expressions are used in the `fn:matches`, `fn:tokenize`, and `fn:replace` functions. Some of the changes are as follows:

- You can no longer match the empty string in a regular expression with `fn:replace` or `fn:tokenize` (you can with `fn:matches`, however). Previously, the empty string in `fn:replace` and `fn:tokenize` matched everything, but in 4.1 it throws an `XDMP-MATCHZERO` exception.
- If you place an invalid escape sequence in a regular expression, an exception is raised. In 4.0, some invalid escape sequences (for example, `\/`) were allowed. In 4.1, any invalid escape sequence throws an exception.
- Certain invalid character classes are no longer allowed. For example, the regular expression `[z-a]` is acceptable in 4.0 (although it does nothing), but throws an exception in 4.1. All invalid character classes now throw an exception.

For example, each of the following expressions contains an invalid regular expression, and throws an exception in 4.1 but completes in 4.0:

```
xquery version "1.0-ml";

fn:replace("", "\s*", "x"),
fn:replace("http://marklogic.com", "\/", "X"),
fn:replace("abc", "[z-a]", "z")
(:
  Throws exception in 4.1, returns the following in 4.0:
  x
  http:XXmarklogic.com
  abc
:)
```

If you have any code that uses these regular expressions, you should review the regular expressions and rewrite it as needed.

4.15.4 Spelling Dictionaries are Now Required to be in the Spell Namespace

When using the spelling correction functions in 4.1, the dictionary documents must be in the `http://marklogic.com/xdmp/spell` namespace. Previously, the dictionary documents were supposed to be in this namespace, but in some cases it still worked if it was in no namespace. If you have dictionaries that are not in the `spell` namespace, then you must transform them to be in that namespace for them to work correctly in 4.1.

4.15.5 `spell:suggest` Now Returns Suggestions Even If Word Is In the Dictionary

The `spell:suggest` function now returns spelling suggestions when a word appears in the dictionary (that is, when it is spelled correctly). In 4.0, `spell:suggest` returns the empty sequence if the word is in the dictionary. To find out if a word is in the dictionary (that is, to find if a word is spelled correctly), use the `spell:is-correct` function. If you have code that relies on the old behavior of `spell:suggest` returning an empty sequence for a correctly spelled word, you should rewrite that code to use `spell:is-correct` in 4.1.

4.15.6 nobody User Has app-user Role

Upgrading to 4.1 adds the `app-user` role to the `nobody` user. The `app-user` role has privileges to run the `xdmp:value` and `xdmp:with-namespaces` functions. If you have any applications that use the `nobody` user, those applications will now run with more privileges than they had in 4.0 (unless you had manually changed the roles assigned to the `nobody` user). The `nobody` user is, by default, the default user for App Servers before authentication takes place. If you do not want the `nobody` user to have these privileges, than you can modify the `nobody` user after the upgrade.

Application Builder relies on the `nobody` user having these privileges, however, so if you modify the `nobody` user (or the `app-user` role), than Application Builder might not work as expected. Additionally, Application Builder uses the `app-user` role to put execute permissions on the error handler, and if the `nobody` user does not have the `app-user` role then no users will be able to log in to applications built with Application Builder.

4.15.7 xdmp:directory-create Throws Exception if Directory Exists

In 4.1, if you try to create a directory (using `xdmp:directory-create`, for example) and the directory already exists, then an exception is thrown (`XDMP-DIREXISTS`). In 4.0, the directory was re-created, and no exception was thrown. If you have code that relies on the old behavior, you must modify that code to make it handle the exception when the directory exists.

4.15.8 XCC Now Requires Java 1.5 or Greater

The 4.1 Java XCC libraries now require Java 1.5 or greater. Previously they required Java 1.4 or greater. If you have XCC applications that run with Java 1.4, in order to use them with a 4.1 XCC package, you will need to upgrade those application environments to use Java 1.5 or greater. If you are still using a 4.0 XCC package, then the applications should continue to work, although you will not be able to use any of the 4.1-specific XCC feature (SSL, for example).

4.15.9 Output Differences in xdmp:query-meters, xdmp:host-status, and xdmp:forest-status Beginning in 4.1-7

Starting with 4.1-7, the `xdmp:query-meters`, `xdmp:host-status`, and `xdmp:forest-status` functions contain additional XML elements in their output. If you have applications that are not expecting these additional elements, it might cause an incompatibility in those application. The only difference in the output is additional elements; all of the information that was there previously is still available. Therefore, an incompatibility would only occur in the event that your application is expecting only the old elements to be returned.

5.0 Other Notes

This section provides the following information about MarkLogic Server:

- [Memory and Disk Space Requirements](#)
- [Compatibility with XQuery Specifications](#)
- [XQuery Extensions](#)
- [Documentation](#)
- [Browser Requirements](#)
- [Support](#)

5.1 Memory and Disk Space Requirements

MarkLogic Server requires at least 512 MB of system memory.

The first time it runs, MarkLogic Server automatically configures itself to the amount of memory on the system, reserving as much as it can for its own use. If you need to change the default configuration, you can manually override these defaults at a later time using the Admin Interface.

MarkLogic recommends the following two guidelines for server sizing:

- Configure your server with 1 GB of physical memory for every 16 GB of source content you expect to manage.
- Configure your server with at least one CPU (or core) per 100 GB of source content.

Pragmatically, we recommend running most configurations with a minimum of two CPUs (or two cores).

MarkLogic Server requires 3 times the disk space of the total forest size. Specifically, each forest on a filesystem requires its filesystem to have at least 3 times the forest size in disk space. This translates to three times the disk space of the source content after it is loaded. For example, if you plan on loading content that will result in a 10 GB database, reserve at least 30 GB of disk space. The disk space reserve is required for merges.

On UNIX systems, MarkLogic Server requires swap space at least equal to the amount of physical memory on the machine. Swap space equal to twice the amount of physical memory is highly recommended. For example, if you have a UNIX machine with 32 GB of memory, you should ideally configure the swap space to be 64 GB (and at least 32 GB). This is true on Windows systems also, but the system is normally set up to grow the swap (page) file as needed.

For more details about memory, disk, and swap requirements, see [Memory, Disk Space, and Swap Space Requirements](#) in the *Installation Guide*.

5.2 Compatibility with XQuery Specifications

This release implements the XQuery language, functions and operators specified in the W3C XQuery 1.0 Recommendations:

- <http://www.w3.org/TR/xquery/>
- <http://http://www.w3.org/TR/xquery-operators/>

Additionally, there is backwards compatibility with the May 2003 version of the XQuery 1.0 Draft specification used in MarkLogic Server 3.2 and previous versions. For details on the XQuery implementation in MarkLogic Server 4.1, including the three different dialects supported, see the *XQuery and XSLT Reference Guide*.

5.3 XQuery Extensions

Working within the W3C XQuery 1.0 Recommendation, MarkLogic has created a number of language extensions enabling key functionality not supported in the current release of the language specification. These extensions provide transactional update capabilities, assorted search and retrieval features, various data manipulation functions, and administrative tools.

The extensions, as well as the XQuery standard functions, are documented at <http://developer.marklogic.com>.

5.4 Documentation

MarkLogic Server includes the following documentation, available through the support web site and through <http://developer.marklogic.com/>:

Documentation	Description
<i>Installation Guide</i>	Provides procedures for installing MarkLogic Server.
<i>Administrator's Guide</i>	Provides procedures for administrative tasks such as creating servers, creating databases, backing up databases, creating users, setting up your security policy, and so on.
<i>Scripting Administrative Tasks Guide</i>	Provides information on writing code to script various administrative tasks such as creating and modifying databases, App Servers, and so on.
<i>Application Developer's Guide</i>	Provides procedures, methodologies, and conceptual information about general MarkLogic Server application development tasks.
<i>Search Developer's Guide</i>	Provides procedures, methodologies, and conceptual information about search-based application development tasks.

Documentation	Description
<i>Application Builder Developer's Guide</i>	Provides step-by step information on using Application Builder to build a search application, and also provides information on customizing applications built with Application Builder.
<i>Content Processing Framework Guide</i>	Provides an introduction to the Content Processing Framework and procedures for installing the default content processing framework.
<i>Understanding and Using Security Guide</i>	Provides information on the role-based security model in MarkLogic Server.
<i>Query Performance and Tuning Guide</i>	Provides performance-related information that is useful to application developers and administrators.
<i>Scalability, Availability, and Failover Guide</i>	Provides information on large-scale system architecture, clustering, availability, and details on setting up shared-disk and local-disk failover.
<i>MarkLogic XQuery and XSLT Function Reference</i>	API documentation for the MarkLogic built-in and module extensions to the XQuery standard functions, as well as API documentation for the W3C functions implemented in MarkLogic Server.
<i>4.1 Release Notes</i>	Contains a summary of new features, upgrade compatible information, and known issues.
XCC Javadoc and .Net C# API Documentation	API documentation for the MarkLogic XML Contentbase Connector for Java API (XCC/J) and .Net XCC C# API documentation.
<i>XCC Developer's Guide</i>	An overview of the what you can do with the XCC libraries, examples of how to use XCC, and an overview of the sample applications included with XCC.
<i>XQuery and XSLT Reference Guide</i>	A condensed overview of the XQuery language, including information on the three XQuery dialects in MarkLogic Server. This book does include some syntax information, although it is primarily intended as in introduction and quick-reference to the language, not as a comprehensive reference.
<i>Getting Started with MarkLogic Server</i>	A quick, step-by-step overview of how to get up and running with MarkLogic Server.

XQuery language documentation is provided through the W3C working group drafts specified in “Compatibility with XQuery Specifications” on page 34. Sample code is provided through the demo server at <http://localhost:8000/>, which is automatically installed as part of the MarkLogic Server installation process. Additionally, there are many samples available on the MarkLogic developer site (<http://developer.marklogic.com>).

XQuery language extensions specific to MarkLogic Server are documented online in the *MarkLogic XQuery and XSLT Function Reference*. Example code snippets are provided as part of that documentation. The Admin Interface provides a large-scale example of complex XQuery programming, using many of the MarkLogic Server XQuery language extensions.

The Admin Interface includes built-in help screens that explain the purpose of the various controls and parameters in the Admin Interface.

Known bugs are documented online as we find them or as they are reported to us. See <http://support.marklogic.com> (supported customers only) for more details.

5.5 Browser Requirements

The MarkLogic Server Admin Interface is supported on Internet Explorer 6 or newer and Firefox browsers.

Application Builder and the applications generated with Application Builder are supported on Internet Explorer 6 or 7 on Windows, Firefox 3.0 or newer on Windows, and Safari 3 or newer on Mac OS 10.5. Other browser/platform combinations may work but are not as thoroughly tested.

Authentication functions in MarkLogic Server are supported for Internet Explorer 6 or newer and Firefox.

5.6 Support

MarkLogic provides technical support according to the terms detailed in your Software License Agreement. For evaluation licenses, MarkLogic may provide support on an “as possible” basis.

For registered customers, we invite you to visit our support website at <http://support.marklogic.com> to access our full suite of documentation and help materials. For all customers, including community licensed customers, visit the MarkLogic Developer’s site at <http://developer.marklogic.com>, which includes full product documentation, downloads, and developer community open-source projects.

If you have questions or comments, you may contact MarkLogic Technical Support at the following email address:

support@marklogic.com

If reporting a query evaluation problem, please be sure to include the sample XQuery code.