

Rocket Software

U2 Technical Support

UniVerse 11 and Beyond

Version 1.3

What You Need to Know Before You Upgrade



Prepared by Jonathan Smith
Lead Technical Support Engineer
1.1 October, 2010, 1.2 September 2011, 1.3 November 2013

© Copyright Rocket Software 2010. All rights reserved. Licensed Materials - Property of Rocket Software.

Overview.....	5
New Memory Structures.....	6
Global Control Table (GCT).....	6
Local Control Table (LCT).....	6
New Daemons.....	7
uvsmm.....	7
uvcleanup.....	8
Clean up process.....	8
Inter-process Communication (IPC) Facility Keys.....	9
U2 Replication.....	10
What is U2 Replication.....	10
Error Messages from U2 Replication.....	10
Replication Daemons.....	11
New UniVerse Configurable Parameters.....	12
Shared Memory Parameters that will require changing.....	12
NUSERS.....	12
SHM_MAX_SIZE.....	12
SHM_GNTBLS.....	12
SHM_GNPAGES.....	12
SHM_GPAGESZ.....	12
Shared Memory Parameters that should not require changing.....	13
SHM_ATT_ADD.....	13
SHM_LBA.....	13
SHM_MIN_ATT.....	13

SHM_LPINENTS	13
SHM_LMINENTS.....	13
SHM_LCINENTS.....	13
SMH_LPAGESZ	13
SHM_FREEPCT.....	13
SHM_NFREES.....	13
NSEM_PSET	13
Calculation Rules / Guidelines for the New Parameters.....	14
NUSERS	14
SHM_GNPAGES	14
SHM_GPAGESZ.....	14
Unix semmni.....	14
Unix semmns	14
Unix shmmni.....	14
Unix semmnu.....	14
Unix semmsl	14
Shared Memory Configuration Error Messages	15
‘No more LCTs’	15
‘No more GCTs’	15
‘uvsmm can’t get the first GSM errno = 22’	15
‘No more shared memory ids’	16
‘No more entries in CI table in LCT-xxx’	16
‘No more entries in MI table in LCT-xxx’	16
‘No more entries in PI table in LCT-xxx’	16

Worked Example	17
U2 Replication Configurable Parameters	18
REP_FLAG.....	18
TCA_SIZE	18
MAX_LRF_FILESIZE.....	18
N_REP_OPEN_FILE	18
MAX_REP_SHMSZ.....	18
REP_LOG_PATH.....	18
MAX_RW_INGRP.....	18
ADE Parameter	18
ADE_OPTIONS	18
UVTSORT	18
UniDebugger.....	19
General Call Interface (GCI)	19
UniVerse Authorization Grace Period.....	19
Starting UniVerse Daemons	20
Stopping UniVerse Daemons.....	20
UniVerse 11.1 ODBC 3.0 Compliancy.....	21
UniVerse 11.1 uvadm functionality.....	22
BUILD.INDEX CONCURRENT.....	23
Connection Pooling.....	23
ADE Enhancements	23

Overview

The introduction of UniVerse 11 could be described as one of the most ambitious releases of UniVerse we ever delivered. Several new major features have been introduced alongside improvements and additions to existing functionality. These include EDA (External Database Access), JPA (Java Persistence API, 64 bit versions), ADE improvements including Encrypted Indexes and Keys, BUILD.INDEX CONCURRENT, Websphere MQ API, Secure HTTP and U2 Replication.

Due to changes to existing features and implementation of new features, there are new daemons, memory structures, configuration parameters and new behaviours that you will see even if you do not use the new functionality. The majority of this document explains these daemons, memory structures, events and their associated configurable parameters. The rest of the document covers some of the other important changes in 11.

This document was first written for UniVerse 11.1 in October 2010 but the document has now been updated for 11.2 in November 2013 and the information contained in it is applicable to all versions of UniVerse 11.1 onwards.

Note: For assistance on shared memory configuration error messages go to section starting on page 15

New Memory Structures

The original UniVerse shared memory segment remains and the new memory structures ported from UniData will be placed after the original memory segment contents.

Global Control Table (GCT)

We ported the Global Control Table (GCT) structure from UniData into UniVerse 11.1 predominantly in order to port the Local Control Table (LCT) structures.

The GCT in UniData is responsible for many other tasks in UniData currently we have only implemented what was needed to implement U2 Replication.

Local Control Table (LCT)

The Local Control Table (LCT) is the system memory table that will now manage login sessions in UniVerse 11.1.

Each uv process will require one slot in the LCT. The number of slots available in the LCT is defined by the tuneable **NUSERS** in uvconfig. We therefore recommend that **NUSERS** be configured appropriately to accommodate your maximum number of simultaneous UniVerse processes.

When calculating this value you need to be aware of how many 'sessions' each user has logged on, the number of phantom processes, the number of WebDE responders, Connection Pools, ODBC connections, UO Connections, UOJ Connections, UO.NET connections etc.

For more information on how to set **NUSERS**, see the "Adminstrating UniVerse 11.1 manual".

If no more slots are available in the LCT an error message will be displayed and logged to uvsmm.errlog file that says 'No more LCTs'.¹

¹ The message appears in the uvsmm.errlog from 11.1.9 on Windows and 11.2.0 on Unix.

New Daemons

A UNIX tool called **showuv** is introduced to show live daemon processes.

```
# showuv
USER  PID  TIME COMMAND
root  409628  0:00 /disk1/unishared/unirpc/unirpcd
root  413728  0:00 /disk1/uv/bin/uvcleanup -config
root  389148  0:00 /disk1/uv/bin/uvdlockd -config
root  331938  0:00 /disk1/uv/bin/uvsmm -t 15
#
```

uvsmm

When UniVerse starts, the **uvsmm** (UniVerse Shared Memory Manager) daemon will setup the initial memory segment. Prior to 11.1 this task was performed by DBsetup. DBSetup will no longer be used at 11.1

uvsmm is also responsible for cleaning up any segments previously allocated to processes that have been marked as dead in the LCT.

Messages related to the normal running of **uvsmm** will be logged to the file **uvsmm.log** in the \$UVHOME directory. Should any errors be encountered by **uvsmm** they will be logged to the file **uvsmm.errlog** in the \$UVHOME directory.

When UniVerse, starts the current contents of all the error logs are appended to a copy of the logs in the \$UVHOME/savedlogs directory. This enables a history of any errors to be kept across multiple restarts of UniVerse.

A new tool to allow a view of the Shared Memory Manager has been introduced called **uvsms**. Its use will be covered in more depth in the formal documentation set.

```
uvsms [options]
```

The options are:

No option: showing GCT & LCT.

-h: showing SMM segment header

-G shm_id: showing a specific shared memory segment

-g shm_nono: showing a specific shared memory segment

-L pid: showing a specific LCT entry

-l lct_no: showing a specific LCT entry

-S shm_id: showing the LCT entry of session where the shared memory segment is created.

uvcleanupd

When UniVerse starts, the uvcleanupd daemon is started alongside uvdlockd. The 'dead' user clean up functionality is now under the control of uvcleanupd prior to 11.1 this was performed by uvdlockd.

The methodology used to detect a 'dead' user process used by uvcleanupd differs to that of uvdlockd.

If a user session is killed, they will have to wait until uvcleanupd has cleaned up the printer segment before they can re-login. Prior to 11.1 the printer memory segment could be 'cleaned up' on login.

Messages related to the normal running of uvcleanupd will be logged to the file **uvcleanupd.log** in the \$UVHOME directory. Should any errors be encountered by uvcleanupd they will be logged to the file **uvcleanupd.errlog** in the \$UVHOME directory.

Clean up process

The cleanup process was moved to uvcleanupd as U2 Replication required that any 'dead' sessions are removed by the system daemon to avoid any unnecessary hangs. The logic of the cleanup is as follows.

- uvcleanupd routinely checks the LCT to find 'dead' user sessions, if found, it marks the entry as dead.
- If a LCT entry is marked as dead, uvcleanupd will cleanup the left-over locks and abort uncommitted replication logs
- If a LCT entry is marked as dead, uvsmm will cleanup the shared memory segments allocated by the dead process
- After all the cleanup processes have been completed, uvcleanupd will remove the printer segment and free the LCT entry for reuse.
- When a foreground user session logs in, it checks whether the printer segment and LCT entry exist:
 - If printer segment does not exist but LCT entry does, it marks it dead.
 - If printer segment exists and previous job leader process dead, it marks it dead.
 - If printer segment exists and previous job leader process is still alive, the login process will wait and prompt a message. Users can force a cleanup by issuing "uvcleanupd -p <process id> -n <signature>" command.

Note: From 11.1, the “`uvdlockd -p`” command does not clean up a printer segment. Instead, the command “`uvcleanupd -p <process id> -n <signature>`” can forcefully mark a LCT entry dead and let the background `uvcleanupd` daemon clean it up. It can be issued by root, the UV admin user or a user with the same signature.

Inter-process Communication (IPC) Facility Keys

With the introduction of the new features in UV11.1, additional system IPC facilities (shared memories, semaphores, and messages) had to be created.

Prior to the introduction of these IPC facilities UniVerse users were used to the use of predefined fixed keys such as “0xacea....” And “0xaceb....”.

UniVerse will now use private keys when an IPC facility is created internally. The IPC IDs are then stored in the UniVerse system main shared memory segment whose key is known by all UniVerse processes.

To assist system administrators, we have provided system level commands *uvipcstat* and *uvipcrm*. As its name suggests, *uvipcstat* will report the IPC facility usage of the UniVerse system while *uvipcrm* will remove the IPC facilities used by the UniVerse system.

U2 Replication

The starting point for U2 Replication was to take UniData Replication from the UniData 7.2 release and port the functionality into UniVerse 11.1. In order to achieve this some of the shared memory structures, IPC (Inter-process communications) and daemons that are present in UniData 7.2 needed to be implemented into UniVerse 11.1.

The implementation of these structures introduces new configurable parameters in uvconfig, new daemons, new memory structures and changes to way that UniVerse handles certain events. It also prepares UniVerse for introduction of the U2 RFS (U2 Recoverable File System) in a latter release.

What is U2 Replication

U2 Replication provides an automated, scalable method to deliver file, record and account level operations together with the data updates and their transactions to other server(s) or other account(s) in Real-Time, Immediate, or Deferred mode, with the added ability for a standby / failover and failback relationship between servers.

By default the files in the replicated environment will be read-only but can be overridden on a file by file basis.

Error Messages from U2 Replication

Traditionally error messages generated by UniVerse contain a number and the text associated with that error from the SYS.MESSAGE file. This has allowed language based customisations of the messages.

Currently at 11.1 and 11.2 the error messages generated by U2 Replication are hard coded into the UniVerse product as this is the method employed in UniData Replication where the initial code was taken from.

This means that currently no customisation of these messages can be performed. The movement to the more traditional SYS.MESSAGE approach will likely happen in a latter release of UniVerse.

For more information see the 'U2 Data Replication Guide' pdf.

Replication Daemons

The following are the extra daemons that are used as part of U2 Replication and are covered in more depth in the documentation and manuals relating to U2 Replication.

- **uvrepmanager** The U2 Replication Daemon. It creates the log files.
- **uvpub** The publisher process of each published replication group
- **uvsub** The subscriber process of each subscribed replication group
- **uvpublistener** The publisher replication listener process
- **uvpubsyncer** The publisher synchronizer process
- **uvrw** Replication Writer processes

New UniVerse Configurable Parameters

There are many new parameters added with the introduction of the UniData shared memory structures, most of them will not require changing from their defaults at 11.1 as they are not yet fully utilized with UniVerse 11.1.

Shared Memory Parameters that will require changing

NUSERS

Limit for number of UniVerse processes that can run at the same time. This will require tuning based upon the number of UniVerse processes that are required to run on the system at the same time.

(Note: If you set NUSERS to a value below the licenced user count, UniVerse will use the licensed user count in place of NUSERS)

SHM_MAX_SIZE

The maximum size (in bytes) of a shared memory segment. This value may need tuning if U2 Replication is in use and the sizes of the Replication Buffers are increased.

SHM_GNTBLS

Number of GCTs (Global Control Tables) in CTL (Control Table List setup by uvsmm). Each shared memory segment is associated with a GCT and **divided** into equal-sized pages called global pages. The GCT registers the use of global pages in its associated shared memory segment.

SHM_GNPAGES

Number of global pages in a shared memory segment.

(Note: SHM_GNPAGES should be a multiple of 32)

SHM_GPAGESZ

Size of each global page, in 512-byte units.

(Note: SHM_GPAGESZ should be a multiple of 32)

Shared Memory Parameters that should not require changing

SHM_ATT_ADD

Starting address for shared memory attachment. Set at installation; do not change this unless instructed by Rocket.

SHM_LBA

Alignment size, in bytes, for shared memory attachment. Set at installation; do not change.

SHM_MIN_ATT

The minimum number of shared memory segments that should be kept attached to a process.

SHM_LPINENTS

Number of entries in the PI table of a LCT, which is the number of processes allowed in a process group (a user session).

SHM_LMINENTS

Number of memory entries in each user session. Each memory entry records a global page or a self-created shared memory segment (in case of a global page is not large enough) allocated to the user session. The default value of 255 does not usually require modification.

SHM_LCINENTS

The number of entries in the CI table of each LCT, which registers how memory pieces recorded in MI entries are used. The default value of 255 does not usually require modification.

SMH_LPAGESZ

Size, in 512-byte blocks, of each local page in a global page. A global page is divided into local pages, so SHM_GPAGESZ must be a multiple of SHM_LPAGESZ.

SHM_FREEPCT

Percentage of freed global pages in an active global shared memory segment that UniVerse keeps in the global shared memory pool. uvsmm checks the current percentage; if the percentage is less than SHM_FREEPCT, uvsmm creates a new shared segment

SHM_NFREES

The number of inactive shared memory segments that UniVerse keeps in the system. uvsmm checks the current number of inactive segments; if the number is larger than SHM_NFREES, uvsmm returns some inactive global shared segments to UNIX.

NSEM_PSET

Number of semaphores per semaphore set.

Calculation Rules / Guidelines for the New Parameters

NUSERS

The value should be set high enough to accommodate the maximum number of simultaneous UniVerse processes that you will require to run on your system. These include UVODBC connections, connection pools, UniObjects connections, Telnet, Phantoms, and Replication Writers etc.

As a general guideline take your licenced user count multiply that by the number of device licences and add 25%.

There is also another rule that if NUSERS is changed $SHM_GNPAGES * SHM_GNTBLS$ has to be equal to or greater than NUSERS.

If you set NUSERS to a value below the licenced user count, UniVerse will use the licensed user count in place of NUSERS.

SHM_GNPAGES

This should be a multiple of 32.

SHM_GPAGESZ

The default is 256.

Unix semmni

The UniVerse minimum requirement is that semmni is at least $(NUSERS / NSEM_PSET) + 1$. For systems with U2 Replication $((NUSERS / NSEM_PSET) * 2) + 10$.

Unix semmns

The UniVerse minimum requirement is that semmns is at least $NUSERS + 10$.

Unix shmmni

The UniVerse minimum requirement is that shmmni must be greater than SHM_GNTBLS and $NUSERS + 5$.

Unix semmnu

The UniVerse minimum requirement is that semmnu is larger than $NUSERS * 3.25$

Unix semmsl

The UniVerse minimum requirement is that semmsl should be greater than or equal to NSEM_PSET.

Note: All these formulas use the assumption of UniVerse being the only application on the server. If other applications are used then the kernel settings may need to be tuned higher.

Shared Memory Configuration Error Messages

The error messages covered in this section are the messages that you could expect to encounter in UniVerse if the configuration parameters in UniVerse and Unix are not high enough to cope with your user load or memory requirements.

'No more LCTs'

Each UniVerse process needs one LCT (Local Control Table) slot. The number of LCTs available is defined by the UniVerse configurable NUSERS.

If the available LCTs are exceeded then a UniVerse process when attempting to start will encounter a message similar to:-

'No more LCTs Process XXXX Exit at ...'

The message is logged to the uvsmm.errlog file ²

To resolve the issue consider increasing the UniVerse configuration parameter NUSERS also check semmnu in UNIX kernel is large enough to accommodate the change.

'No more GCTs'

If the available GCTs (Global Control Tables) are exceeded then a UniVerse process will encounter a message similar to:-

'No more GCTs Process XXXX Exit at ...'

The message will also be logged to the uvsmm.errlog file.

This means that you have already used as many segments as your system currently supports. To resolve the issue consider increasing shmmni in the UNIX kernel, or increasing the UniVerse configuration parameter SHM_GNTBLS.

'uvsmm can't get the first GSM errno = 22'

uvsmm cannot acquire the first shared memory segment to build the necessary control tables because shmmax is not large enough. Increase the kernel parameter shmmax.

The message will also be logged to the uvsmm.errlog file.

² The message appears in the uvsmm.errlog from 11.1.9 on Windows and 11.2.0 on Unix

'No more shared memory ids'

You are out of shared memory ids. Adjust the UNIX kernel parameter `shmmni` to increase the limit.

The message will also be logged to the `uvsmm.errlog` file.

The error messages covered in this section are the messages that you could encounter in UniVerse but have not been seen to date and are extremely unlikely to occur.

'No more entries in CI table in LCT-xxx'

The CI table in the specified LCT is full. A process has used its limit of local sections. A local section is a local page or several contiguous local pages. Consider increasing the UniVerse configuration parameter `SHM_LCINENTS`.

The message will also be logged to the `uvsmm.errlog` file.

'No more entries in MI table in LCT-xxx'

The MI table in the specified LCT is full. A process has used its limit of global pages. Consider increasing the size of a global page (`SHM_GPAGESZ`) or the number of global pages per process (`SHM_LMINENTS`).

The message will also be logged to the `uvsmm.errlog` file.

'No more entries in PI table in LCT-xxx'

The PI table in the specified LCT is full. Your application has too many forked processes. Your application may not be structured in the correct manner. Consider increasing the UniVerse configuration parameter `SHM_LPINENTS`.

The message will also be logged to the `uvsmm.errlog` file.

Worked Example

For example an enterprise licence of 150 users should have NUSERS set to 1875 $(150 * 10) + 25\%$. (Note – The example here is assuming a maximum of 10 connections for each user, this value should be adjusted to your ‘average’ number of connections per user).

Change SHM_GNTBLS to 32 and change SHM_GNPAGES to 64 this ensures $32 * 64 = 2048$ which is greater than NUSERS.

Check the Unix value of semmni is at least 1232. $(NUSERS / (NSEM_PSET+1)) + 1024$.

Check the Unix value of semmns is at least 3933. $(NUSERS + 10 + 2048)$

Check the Unix value of shmmni is at least 1880. (Greater of SHM_GNTBLS or NUSERS +5)

Check the Unix value of semmnu is least 1775 (NUSERS).

U2 Replication Configurable Parameters

REP_FLAG

Turns U2 Replication on or off. If this value is “0” then U2 Replication is off. If this value is a positive integer, it is on.

TCA_SIZE

The maximum number of entries in the Transaction Control Area.

MAX_LRF_FILESIZE

The maximum Log Reserve File size, in bytes.

N_REP_OPEN_FILE

The maximum number of U2 replication related O/S files that a uvsh process can have open at one time.

MAX_REP_SHMSZ

The maximum shared memory buffer segment size - this should not exceed SHM_MAX_SIZE

REP_LOG_PATH

Full path to the U2 replication log files.

MAX_RW_INGRP

The maximum number of RWs (Replication Writers) in a Replication Group.

ADE Parameter

ADE_OPTIONS

A “bit-wise” flag in order to control behaviour of ADE functions in future releases. Currently set to “0” and reserved for use on future releases.

For more information see the ‘Administrating UniVerse’ and ‘UniVerse System Description’ pdf’s.

UVTSORT

We now recommend that UVTSORT should be disabled and is automatically disabled from UniVerse from release 11.1.9 onwards.

UniDebugger

UniDebugger has been superseded by U2 Basic Development Toolkit (BDT) and has been removed from the UniVerse client CD.

General Call Interface (GCI)

As with any upgrade of UniVerse, existing GCI routines will need to be re-linked into UniVerse and a new uvsh (UniVerse Shell) process created.

In the 11.1 HPUX Itanium and AIX UniVerse versions a significant change has been made in that the uvsh is now a 64 bit executable, previous versions were a 32 bit executable. As UniVerse 11 is further developed more 64 bit ports will be becoming available.

This means that you need to ensure that if you use GCI on these releases you have a C compiler and linker capable of compiling and linking 64 bit applications on that platform and that your GCI libraries are compiled as 64 bit modules.

In addition, any third party libraries or operating system interfaces that are directly or indirectly referenced by your GCI definitions or libraries, regardless of whether statically or dynamically linked with uvsh, must be 64 bit libraries or interfaces.

Special attention should be paid to the data types used in your GCI modules, especially function arguments and result types, as assumptions regarding the sizes of the C language int, long, and pointer data types may no longer be valid.

The information on which compiler has been used to build UniVerse on a given platform is included with the release notes and available on the Product Availability Matrix (PAM) at <https://u2tc.rocketsoftware.com/matrix.asp>.

For more information see the 'UniVerse GCI Guide' pdf

UniVerse Authorization Grace Period

Prior versions of UniVerse would run for a grace period of 60 periods before requiring authorization. This period has been reduced to 10 days in 11.1 and in version 10.3.9 and higher of UniVerse.

For more information see the 'Adminstrating UniVerse' pdf

Starting UniVerse Daemons

Use one of the following commands to start the UniVerse daemons:

- `uv -admin -start [-init]`
- `uv.rc start [init]`

If you specify the `-init` option, U2 Data Replication clears the pending replication logs saved to synchronize the subscriber database. You can use this option for initial startup, or after refreshing a subscriber database.

Stopping UniVerse Daemons

Use one of the following commands to stop the UniVerse daemons:

- `uv -admin -stop [-force]`
- `uv.rc stop [force]`

These commands check whether any users are still logged on to the system. If users are logged on, UniVerse does not stop the system. If you specify the `-force` option, UniVerse stops the system even if there are users logged on to the system.

UniVerse 11.1 ODBC 3.0 Compliancy

Changes have been made to make UniVerse 11.1 and the 11.1 UVODBC driver ODBC 3.0 compliant and also to ensure the driver is “thread safe”. In addition we now also support the SQL UNNEST statement and have implemented 21 additional ODBC 3.0 API functions.

The clients install is now ‘Rocket’ branded and will now install under ‘Rocket’ directories and ‘Rocket’ registry keys.

Note: The UniVerse 11.1 engine has changed from a 32 bit engine to a 64 bit engine for the HPUX Itanium and AIX releases (more will follow). If you have been using BCI on these platforms you will need to update your UNIX odbc driver from 32 bit to 64 bit in order for BCI to continue to work.

For more information see the ‘UniVerse ODBC Guide’ and ‘UniVerse BASIC SQLClient Interface Guide (BCI)’ pdfs

UniVerse 11.1 uvadm functionality

UniVerse 11.1 has been enhanced to allow the 'uvadm' user or members of the 'uvadm' group to perform many of the administration tasks in UniVerse that would have previously required a 'root' user login.

A new command 'uv_upgrade' has been added to allow the upgrade of UniVerse to be performed as the 'uvadm' user and not needing a root login.

The basic programs in the SYSADM.MENU such as configure spooler, device etc. have been modified to now be performed by the 'uvadm' user.

The 'uvadm' user can now also perform the following commands at 11.1:-cleanup_lock

- convchar
- convencfile
- encman
- ld_nls_map
- list_readu (LIST.READU)
- load_shm_cat
- lock_maint
- master
- message
- modify_shm
- resize
- udpaccount
- unlock (UNLOCK)
- uvbackup/uvrestore
- uvdrlogd
- uvdrrepdm
- uvfixfile
- uvlictool
- uvregen
- uvtic
- uvtide

Note: uv.rc should not be used directly unless you are root. Instead use \$UVBIN/uv -admin -start or -stop. \$UVBIN/uvadmsh should be owned by root, and have sticky bit on (mode should be 04751 or 04755)

For more information see the 'Adminstrating Universe' and 'Installing UniVerse' pdfs

BUILD.INDEX CONCURRENT

UniVerse 11.1 now allows an alternate key index to be built while the file is in use with the addition of the CONCURRENT keyword to the BUILD.INDEX command. Prior to 11.1 exclusive access was needed to a file to build an index.

The ability to build an index while still using the file reduces the necessity for downtime and further aids support of the 24-7 environments we are all now required to operate in.³

For more information see the 'UniVerse Command Reference' pdf.

Connection Pooling

Although not exclusive to 11.1 the 11.1 version of the clients includes versions 4.1.4 of UOJ.dll and 2.2.4 of UODOTNET.dll. At these versions the behaviour of the minimum pool size has been changed.

Prior to these versions the minimum pool size could be set to "0" but the effective minimum pool size was "1" as the pool was never allowed to shrink below a size of one. Now if the minimum pool size is set to "0", the pool will shrink to zero and release all of the server side connection pool licences it was holding.

For more information see the 'UniObjects for Java Developer's Guide' and the 'UniObjects for .NET Developer's Guide' pdf's.

ADE Enhancements

Automatic data encryption has undergone several improvements for 11.1. Two of these are that the key to a file and an index can now be encrypted.

For more information see the 'UniVerse Security' pdf.

³ When creating a new index or disabling / enabling existing indexes this action will not be recognized by other users or processes (e.g. uvrw) that already had the file open. They will need to close and reopen the file to do so.