Rocket UniVerse 11 Structural Changes

What you need to know before you install or upgrade to UniVerse 11

Rocket U2 Technical Support Supplemental Information

April 2014

UNV-112-REP-OG-1
Notices

Edition

Publication date: April 2014
Book number: UNV-112-REP-OG-1
Product version: Rocket UniVerse V11.2

Copyright

© Rocket Software, Inc. or its affiliate 1985-2014. All Rights Reserved.

Trademarks

Rocket is a registered trademark of Rocket Software, Inc. For a list of Rocket registered trademarks go to: www.rocketsoftware.com/about/legal. All other products or services mentioned in this document may be covered by the trademarks, service marks, or product names of their respective owners.

Examples

This information might contain examples of data and reports. The examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

License agreement

This software and the associated documentation are proprietary and confidential to Rocket Software, Inc. or its affiliates, are furnished under license, and may be used and copied only in accordance with the terms of such license.

Note: This product may contain encryption technology. Many countries prohibit or restrict the use, import, or export of encryption technologies, and current use, import, and export regulations should be followed when exporting this product.

Contact information

Website: www.rocketsoftware.com
Rocket Software, Inc. Headquarters
77 4th Avenue, Suite 100
Waltham, MA 02451-1468
USA
Tel: +1 781 577 4321
Fax: +1 617 630 7100
Contacting Global Technical Support

If you have current support and maintenance agreements with Rocket Software, you can access the Rocket Customer Portal to report and track a problem, to submit an enhancement request or question, or to find answers in the U2 Knowledgebase. The Rocket Customer Portal is the primary method of obtaining support.

To log in to the Rocket Customer Portal, go to:

www.rocketsoftware.com/support

If you do not already have a Rocket Customer Portal account, you can request one by clicking Need an Account? on the Rocket Customer Portal login page.

Alternatively, you can contact Global Technical Support by email or by telephone:

Email: u2support@rocketsoftware.com
Telephone:
North America +1 800 729 3553
United Kingdom/France +44 (0) 800 773 771 or +44 (0) 20 8867 3691
Europe/Africa +44 (0) 20 8867 3692
Australia +1 800 707 703 or +61 (0) 29412 5450
New Zealand +0800 505 515
## Contents

Notices .............................................................................................................................................................................. 2

Contacting Global Technical Support ................................................................................................................................. 3

Contacts ................................................................................................................................................................................ 4

Chapter 1: Overview ............................................................................................................................................................. 6

Chapter 2: New Memory Structures ....................................................................................................................................... 7

Global Control Table (GCT) .................................................................................................................................................... 7

Local Control Table (LCT) ...................................................................................................................................................... 7

Chapter 3: New Daemons ...................................................................................................................................................... 8

uvsmm daemon ........................................................................................................................................................................ 8

 uvsms [options] .................................................................................................................................................................. 9

uvecleanup daemon ............................................................................................................................................................. 9

Cleanup process ................................................................................................................................................................. 10

Chapter 4: Inter-process Communication (IPC) Facility Keys ............................................................................................... 11

U2 Replication ...................................................................................................................................................................... 12

What is U2 Replication? ...................................................................................................................................................... 12

Error Messages from U2 Replication ................................................................................................................................ 12

Replication Daemons .......................................................................................................................................................... 13

Chapter 5: New UniVerse Configurable Parameters ........................................................................................................ 14

Shared Memory Configuration Error Messages .................................................................................................................... 17

‘No more LCTs’ .................................................................................................................................................................... 17

‘No more GCTs’ .................................................................................................................................................................... 17

‘uvsmm can’t get the first GSM errno = 22’ ...................................................................................................................... 17

‘No more shared memory IDs’ ........................................................................................................................................... 18

‘No more entries in CI table in LCT-xxx’ .......................................................................................................................... 18

‘No more entries in MI table in LCT-xxx’ .......................................................................................................................... 18

‘No more entries in PI table in LCT-xxx’ .......................................................................................................................... 18

Worked Example ................................................................................................................................................................. 19

U2 Replication Configurable Parameters ............................................................................................................................... 20

ADE Parameter ...................................................................................................................................................................... 20

ADE_OPTIONS ...................................................................................................................................................................... 20

UVTSORT ................................................................................................................................................................................... 20

Chapter 6: UniDebugger ........................................................................................................................................................... 21

General Call Interface (GCI) ................................................................................................................................................. 21

UniVerse Authorization Grace Period ................................................................................................................................... 21
Chapter 1: Overview

The introduction of UniVerse 11 could be described as one of the most ambitious releases of UniVerse we have 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.

Because of changes to existing features and implementation of new features, there are new daemons, memory structures, configuration parameters, and new behaviors that you will see even if you do not use the new functionality. This document explains these daemons, memory structures, events, and the associated configurable parameters, as well as other important changes in version 11.1 and 11.2.

**Note:** For assistance on shared memory configuration error messages, refer to the [Shared Memory topic](#).

To access the Rocket UniVerse documentation, including the documentation that is referenced in this document, go to [http://www.rocketsoftware.com/brand/rocket-u2/technical-documentation](http://www.rocketsoftware.com/brand/rocket-u2/technical-documentation).
Chapter 2: 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 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 in the LCT is defined by the tunable 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 is logged in to, the number of phantom processes, the number of WebDE responders, connection pools, ODBC connections, UO connections, UOJ connections, and UO.NET connections.

For more information on how to set NUSERS, see Administering UniVerse on UNIX Platforms and Administering UniVerse on Windows Platforms.

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

1 The message appears in the uvsmm.errlog file in version 11.1.9 on Windows and version 11.2.0 on UNIX.
Chapter 3: New Daemons

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

```
# showuv

<table>
<thead>
<tr>
<th>USER</th>
<th>PID</th>
<th>TIME</th>
<th>COMMAND</th>
</tr>
</thead>
<tbody>
<tr>
<td>root</td>
<td>409628</td>
<td>0:00</td>
<td>/disk1/unishared/unirpc/unirpcd</td>
</tr>
<tr>
<td>root</td>
<td>413728</td>
<td>0:00</td>
<td>/disk1/uv/bin/uvcleanupd -config</td>
</tr>
<tr>
<td>root</td>
<td>389148</td>
<td>0:00</td>
<td>/disk1/uv/bin/uvdlockd -config</td>
</tr>
<tr>
<td>root</td>
<td>331938</td>
<td>0:00</td>
<td>/disk1/uv/bin/uvsmm -t 15</td>
</tr>
</tbody>
</table>
```

uvsmm daemon

When UniVerse starts, the uvsmm (UniVerse Shared Memory Manager) daemon will set up the initial memory segment. Prior to version 11.1, this task was performed by DBsetup. DBSetup will no longer be used beginning with version 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 `uvsmm.log` file in the `$UVHOME` directory. If any errors are encountered by uvsmm, they will be logged to the `uvsmm.errlog` file in the `$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. Refer to *UniVerse User Reference* for more information.
### uvsms [options]

<table>
<thead>
<tr>
<th>uvsms option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>No option</td>
<td>Shows GCT &amp; LCT</td>
</tr>
<tr>
<td>-h</td>
<td>Shows SMM segment header</td>
</tr>
<tr>
<td>-G shm_id</td>
<td>Shows a specific shared memory segment</td>
</tr>
<tr>
<td>-g shm_nono</td>
<td>Shows a specific shared memory segment</td>
</tr>
<tr>
<td>-L pid</td>
<td>Shows a specific LCT entry</td>
</tr>
<tr>
<td>-l lct_no</td>
<td>Shows a specific LCT entry</td>
</tr>
<tr>
<td>-S shm_id</td>
<td>Shows the LCT entry of session where the shared memory segment is created</td>
</tr>
</tbody>
</table>

### uvccleanupd daemon

When UniVerse starts, the uvccleanupd daemon is started alongside uvdlockd. The 'dead' user cleanup functionality is now under the control of uvccleanupd. Prior to version 11.1, cleanup was performed by uvdlockd.

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

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

Messages related to the normal running of uvccleanupd will be logged to the `uvccleanupd.log` file in the `$UVHOME` directory. If any errors are encountered by uvccleanupd they will be logged to the `uvccleanupd.errlog` file in the `$UVHOME` directory.
Cleanup process

The cleanup process was moved to uvcleanupd because U2 Replication requires that any 'dead' sessions be 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 any are found, it marks the entry as dead.
- If an LCT entry is marked as dead, uvcleanupd will clean up the leftover locks and terminate uncommitted replication logs.
- If an LCT entry is marked as dead, uvsmm will clean up 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 the printer segment does not exist but an LCT entry does, it marks it dead.
  - If the printer segment exists and the previous job leader process is dead, it marks it dead.
  - If the printer segment exists and the previous job leader process is still alive, the login process will wait and prompt a message. Users can force a clean-up by issuing the `uvcleanupd -p <process id> -n <signature>` command.

Note: Beginning with version 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 an LCT entry as 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.
Chapter 4: Inter-process Communication (IPC) Facility Keys

With the introduction of the new features in version 11.1, 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 the 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 change, some of the shared memory structures, IPC (inter-process communication) and daemons that are 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 the way that UniVerse handles certain events.

Note: Please read the statement at the end of this document on Disaster Recovery.

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 customizations of the messages.

In versions 11.1 and 11.2, the error messages generated by U2 Replication are hardcoded into the UniVerse product because this is the method employed in UniData Replication from which the initial code was taken.

Currently no customization of these messages can be performed.

For more information see U2 Data Replication for Rocket UniVerse.
Replication Daemons

The following extra daemons are used as part of U2 Replication and are covered in more depth in the documentation about U2 Replication.

<table>
<thead>
<tr>
<th>Replication daemon</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>uvrepmanager</td>
<td>The U2 Replication Daemon. It creates the log files</td>
</tr>
<tr>
<td>uvpub</td>
<td>The publisher process of each published replication group</td>
</tr>
<tr>
<td>uvsub</td>
<td>The subscriber process of each subscribed replication group</td>
</tr>
<tr>
<td>uvpublishener</td>
<td>The publisher replication listener process</td>
</tr>
<tr>
<td>uvpubsyncer</td>
<td>The publisher synchronizer process</td>
</tr>
<tr>
<td>uvrw</td>
<td>Replication Writer processes</td>
</tr>
</tbody>
</table>
Chapter 5: New UniVerse Configurable Parameters

Many new parameters were added with the introduction of the UniData shared memory structures. Most of them will not require changing from their defaults at 11.1 because they are not yet fully utilized with UniVerse 11.1.

Table 1: Shared Memory Parameters that will require changing

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NUSERS</td>
<td>Limit for the number of UniVerse processes that can run at the same time. This will require tuning based on the number of UniVerse processes that are required to run on the system at the same time. <em>(Note: If you set NUSERS to a value below the licensed user count, UniVerse will use the licensed user count in place of NUSERS.)</em></td>
</tr>
<tr>
<td>SHM_MAX_SIZE</td>
<td>The maximum size (in bytes) of a shared memory segment. This value might need tuning if U2 Replication is in use and the sizes of the replication buffers are increased.</td>
</tr>
<tr>
<td>SHM_GNTBLS</td>
<td>Number of GCTs (Global Control Tables) in CTL (Control Table List set up 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.</td>
</tr>
<tr>
<td>SHM_GNPAGES</td>
<td>Number of global pages in a shared memory segment. <em>(Note: SHM_GNPAGES should be a multiple of 32.)</em></td>
</tr>
<tr>
<td>SHM_GPAGESZ</td>
<td>Size of each global page, in 512-byte blocks. <em>(Note: SHM_GPAGESZ should be a multiple of 32.)</em></td>
</tr>
</tbody>
</table>
Table 2: Shared Memory Parameters that should not require changing

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NSEM_PSET</td>
<td>Number of semaphores per semaphore set.</td>
</tr>
<tr>
<td>SHM_ATT_ADD</td>
<td>Starting address for shared memory attachment. Set at installation; do not change this setting unless instructed by Rocket Technical Support.</td>
</tr>
<tr>
<td>SHM_FREEPCT</td>
<td>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.</td>
</tr>
<tr>
<td>SHM_LBA</td>
<td>Alignment size, in bytes, for shared memory attachment. Set at installation; do not change.</td>
</tr>
<tr>
<td>SHM_LCINENTS</td>
<td>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.</td>
</tr>
<tr>
<td>SHM_LMINENTS</td>
<td>Number of memory entries in each user session. Each memory entry records a global page or a self-created shared memory segment (in case a global page is not large enough) allocated to the user session. The default value of 255 does not usually require modification.</td>
</tr>
<tr>
<td>SMH_LPAGESZ</td>
<td>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.</td>
</tr>
<tr>
<td>SHM_LPINENTS</td>
<td>Number of entries in the PI table of an LCT, which is the number of processes allowed in a process group (a user session).</td>
</tr>
<tr>
<td>SHM_MIN_ATT</td>
<td>Minimum number of shared memory segments that should be kept attached to a process.</td>
</tr>
<tr>
<td>SHM_NFREES</td>
<td>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.</td>
</tr>
</tbody>
</table>
### Table 3: Calculation Rules / Guidelines for the New Parameters

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Description</th>
</tr>
</thead>
</table>
| 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, multiple your licensed user count by the number of device licenses and add 25%.  
There is also another rule that if NUSERS is changed, SHM_GNPAGES * SHM_GNTBLS must be equal to or greater than NUSERS.  
If you set NUSERS to a value below the licensed 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 must be at least (NUSERS / NSEM_PSET) + 1. For systems with U2 Replication it should be least ((NUSERS / NSEM_PSET) * 2) + 10.                                        |
| Unix semmns    | The UniVerse minimum requirement is that semmns must be at least NUSERS + 10.                                                                                                                               |
| Unix semmnu    | The UniVerse minimum requirement is that semmnu must be larger than NUSERS * 3.25.                                                                                                                         |
| Unix semmsl    | The UniVerse minimum requirement is that semmsl should be greater than or equal to NSEM_PSET.                                                                                                               |
| Unix shmmni    | The UniVerse minimum requirement is that shmmni must be greater than SHM_GNTBLS and NUSERS +5.                                                                                                               |

**Note:** All these formulas assume that UniVerse is the only application on the server. If other applications are used, the kernel settings might 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 for 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, when a UniVerse process attempts to start, it will encounter a message similar to:

‘No more LCTs Process XXXX Exit at …’

The message is logged to the uvsmm.errlog file.

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

‘No more GCTs’

If the available GCTs (Global Control Tables) are exceeded, 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’

2 The message appears in the uvsmm.errlog file in version 11.1.9 on Windows and version 11.2.0 on UNIX.
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.

‘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.

Uncommon error messages

The following error messages might occur in UniVerse, but they 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 might not be structured correctly. 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 license of 150 users should have NUSERS set to 1875 \((150 \times 10) + 25\%\). *(Note:* The example assumes a maximum of 10 connections for each user. Adjust this value to your ‘average’ number of connections per user.)*

- Change `SHM_GNTBLS` to 32 and change `SHM_GNPAGES` to 64 to ensure \(32 \times 64 = 2048\), which is greater than NUSERS.
- Check that the UNIX value of `semmni` is at least 1232. \((\text{NUSERS} / (\text{NSEM}_\text{PSET}+1)) + 1024\).
- Check that the UNIX value of `semmns` is at least 3933. \((\text{NUSERS} + 10 + 2048)\).
- Check that the UNIX value of `shmmni` is at least 1880. \(\text{Greater of SHM_GNTBLS or NUSERS +5}\).
- Check that the UNIX value of `semmnu` is at least 1775 \((\text{NUSERS})\).
U2 Replication Configurable Parameters

Table 4: Configurable parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>REP_FLAG</td>
<td>Turns U2 Replication on or off. If this value is “0,” U2 Replication is off. If this value is a positive integer, it is on.</td>
</tr>
<tr>
<td>TCA_SIZE</td>
<td>The maximum number of entries in the Transaction Control Area.</td>
</tr>
<tr>
<td>MAX_LRF_FILESIZE</td>
<td>The maximum Log Reserve File size, in bytes.</td>
</tr>
<tr>
<td>N_REP_OPEN_FILE</td>
<td>The maximum number of U2 replication-related O/S files that a uvsh process can have open at one time.</td>
</tr>
<tr>
<td>MAX_REP_SHMSZ</td>
<td>The maximum shared memory buffer segment size, which should not exceed SHM_MAX_SIZE.</td>
</tr>
<tr>
<td>REP_LOG_PATH</td>
<td>Full path to the U2 replication log files.</td>
</tr>
<tr>
<td>MAX_RW_INGRP</td>
<td>The maximum number of RWs (Replication Writers) in a Replication Group.</td>
</tr>
</tbody>
</table>

ADE Parameter

ADE_OPTIONS

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

For more information, see Administering UniVerse and UniVerse System Description.

UVTSORT

We now recommend that UVTSORT should be disabled. It is automatically disabled beginning with UniVerse 11.1.9.
Chapter 6: 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 HP-UX Itanium and AIX UniVerse versions, a significant change has been made in that the uvsh is now a 64-bit executable. Previous versions were 32-bit executables. As UniVerse 11 is further developed, more 64-bit ports will become 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, because assumptions regarding the sizes of the C language int, long, and pointer data types might no longer be valid.

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

For more information, see the Rocket UniVerse GCI Guide.

UniVerse Authorization Grace Period

Prior versions of UniVerse ran for a grace period of 60 days before requiring authorization. This period has been reduced to 10 days in version 10.3.9 and later of UniVerse.

For more information, see Administering Rocket UniVerse on UNIX Platforms and Administering Rocket UniVerse on Windows Platforms.
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 users are logged on to the system.

UniVerse 11.1 ODBC 3.0 Compliance

Changes have been made to make UniVerse 11.1 and the 11.1 UVODBC driver ODBC 3.0-compliant and also to ensure that 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 installation 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 HP-UX Itanium and AIX releases. 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 Rocket U2 ODBC Developer’s Guide and Rocket UniVerse BASIC SQL Client Interface Guide (BCI).
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 does not require 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
- uvdrlgd
- uvdrrepdm
- uvfixfile
- uvlictool
- uvregen
- uvtic
- uvtidc

**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 *Administering Rocket UniVerse on UNIX Platforms, Administering Rocket UniVerse on Windows Platforms*, and *Rocket UniVerse Installation Guide*. 
**BUILD.INDEX CONCURRENT**

UniVerse 11.1 now allows an alternative 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 version 11.1, exclusive access to a file was needed 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 24x7 environments we are all now required to operate in.³

For more information, see the *UniVerse User Reference*.

**Connection Pooling**

Although not exclusive to version 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 behavior 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” because 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 licenses it was holding.

For more information, see the *UniObjects for Java Developer’s Guide* and the *UniObjects for .NET Developer’s Guide*.

**ADE Enhancements**

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

For more information, see *UniVerse Security Features*.

³ When you create an index or disable or enable existing indexes, this action will not be recognized by other users or processes (for example, uvrw) that already had the file open. Those processes will need to close and reopen the file to do so.
Disaster Recovery Notice

Designing, building, implementing and maintaining your Disaster Recovery solution is very complex, including server, firmware, storage, and networking. Configuring and implementing replication to meet your Disaster Recovery Service Level Agreements (SLAs) and requirements, while considering the performance and current SLAs in your production environment requires planning and experience. U2 Replication is not a product where you should “Teach Yourself”. Disaster Recovery is not a place to “Learn on the Job”. Inadvertent omissions and mistakes during implementation may have severe consequences for your data and your organization.

Please include Rocket U2 Education and Rocket MV Professional Services in your plan for success.

There are five products Rocket Software does not recommend that you implement by yourself without Rocket U2 Education for your staff or assistance Rocket MV Professional Services:

1. Automatic Data Encryption (ADE)
2. Recoverable File System (RFS)
3. U2 Audit Logging
4. U2 Replication
5. Web/DE

Rocket U2 Education supplies standard courses to learn U2 Replication. These should be included in your overall plan for success.

We “Highly” recommend you rely upon the experience gained by many installations that Rocket MV Professional Services has performed.

Rocket MV Professional Services offers standard packages of services to assist you in implementing U2 Replication which includes methodologies, project plans, implementation assistance, knowledge transfer to your staff, post “Go Live” assistance, and the “As Built” Document defining how your system was implemented which includes recovery procedures.

The “As Built” Document is distributed to your staff, your Application Provider (if applicable) and the Rocket U2 Support organization. These multiple layers of support all will assist you in getting your system back up when failure occurs. They all need to be informed as to what was implemented.

Should you still wish to attempt this yourself (not recommended), Rocket MV Professional Services offers a “Validation Package” where we review what you have implemented and make recommendations for an effective implementation.

Restating:

Please include Rocket U2 Education and Rocket MV Professional Services in your plan for success.