

VSI OpenVMS

ServiceControl V3.5

Release Notes

March 2019

Revision/Update Information
Software Version
Operating System Version

Updated manual.
VSI OpenVMS ServiceControl V3.5
OpenVMS Alpha V7.3-2 & higher
OpenVMS I64 V8.2 & higher



March 2019

Copyright © 2019 VMS Software, Inc., (VSI), Bolton Massachusetts, USA.

VMS Software Inc. makes no warranty of any kind with regard to this material, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. VMS Software Inc. shall not be liable for errors contained herein or for incidental or consequential damages in connection with the furnishing, performance, or use of this material.

This document contains proprietary information, which is protected by copyright. No part of this document may be photocopied, reproduced, or translated into another language without the prior written consent of VMS Software Inc. The information contained in this document is subject to change without notice

HPE, the HPE logo, and OpenVMS are trademarks of Hewlett-Packard Enterprise.

Microsoft, MS-DOS, Windows, and Windows NT are trademarks of Microsoft Corporation in the U.S. and/or other countries.

All other product names mentioned herein may be trademarks of their respective companies.

Confidential computer software. Valid license from VSI required for possession, use or copying.

VMS Software Inc. shall not be liable for technical or editorial errors or omissions contained herein. The information is provided “as is” without warranty of any kind and is subject to change without notice. The warranties for VMS Software Inc. products are set forth in the express limited warranty statements accompanying such products. Nothing herein should be construed as constituting an additional warranty.

Contents

Preface	5
Installation Release Notes	6
1.1 Upgrade Path	6
1.2 Installing VSI OpenVMS ServiceControl.....	6
1.3 Upgrading VSI OpenVMS ServiceControl	7
1.4 Upgrade post installation tasks.....	7
1.4.1 Convert existing OSC configuration databases.....	7
1.4.2 Increase SYSTEM account quota settings	7
Supported Versions	9
2.1 Advisory for running VSI OpenVMS ServiceControl on OpenVMS V7.3-2..	9
2.2 OSC Management GUI advisory	10
New Features.....	11
3.1 OSC agents.....	12
3.1.1 New resource attribute ScriptExecUser – User to run DCL script action routines can be defined resource specific.....	12
3.2 OSC management utility (OSC\$MGR).....	13
3.2.1 Parameter list and wildcard support	13
3.2.2 New SHOW TRANSACTION command	14
3.2.3 New CANCEL TRANSACTION command	15
3.2.4 New Category column in SHOW SRVGRP output.....	18
3.2.5 New Critical column in SHOW RESOURCE output.....	18
3.3 OSC configuration utility (OSC\$CFG).....	20
3.3.1 Attribute data field extension	20
3.3.2 OSC configuration databases are automatically converted to new format.	20
3.3.3 New /ATTRIBUTE qualifier for the MODIFY command.....	21
3.3.4 New COPY SRVGRP command.....	21
3.3.5 New COPY SERVICE command.....	22
3.3.6 New COPY RESOURCE command	22
3.3.7 New RENAME SRVGRP command	23
3.3.8 New RENAME SERVICE command	24
3.3.9 New RENAME RESOURCE command.....	25
3.4 OSC GUI connectivity server.....	27
3.5 OSC event notification.....	28
3.5.1 OSC entity type passed in P8 to DCL user event scripts.....	28
Bug Fixes	29
4.1 OSC master control engine (OSC\$CTRL)	30
4.1.1 Transaction does not terminate if an OSC service engine is restarted or fails while executing this transaction	30

4.1.2	Critical resource of a failover group fails during initial OSC cluster state transition.....	30
4.1.3	Starting the same transaction from different OSC consoles.....	31
4.1.4	ONLINE RESOURCE/NODE fails if the resource is a member of a Parallel or MultilInstance Service Group and the Service Group is already online on a node.....	31
4.2	OSC service engine (OSC\$SRV).....	32
4.2.1	If an ONLINE transaction fails the transaction was not removed from the transaction queue.....	32
4.2.2	If an OFFLINE transaction fails the transaction was not removed from the transaction queue.....	32
4.2.3	Invalid resource state reported to OSC master control if an OFFLINE action routine fails	32
4.2.4	ONLINE transaction fails if a non-critical resource fails to go online	33
4.2.5	Transaction does not terminate if an OSC agent is restarted or fails while executing an action routine.....	33
4.4	OSC agents.....	34
4.4.1	OnlineRetryLimit is set to 0 if a resource fails to go online.....	34
4.5	OSC management utility (OSC\$MGR).....	35
4.5.1	ONLINE transactions on failover Service Groups return wrong status code	35
4.5.2	ONLINE transactions on parallel Service Groups return to the caller before the Service Groups are online on the all CSC cluster members	35
4.5.3	ONLINE transactions on non-critical resources return wrong status code if the transaction fails.....	36
4.5.4	/NODE qualifier is ignored in the ONLINE SERVICE command	36
4.5.5	SHOW CLUSTER returns SS\$_INVARG when called from the command line	36
4.5.6	SHOW VERSION returns SS\$_INVARG when called from the command line	37
4.5.7	No output displayed when SHOW RESOURCE, SHOW SERVICE or SHOW SRVGRP called from the command line.....	37
4.6	OSC configuration utility (OSC\$CFG).....	38
4.6.1	Loading OSC configuration template definition deletes OSC cluster configuration.....	38
4.6.2	Votes = 1 is not automatically assigned to all cluster members.....	38

Preface

Audience

This manual is intended for all system managers who have installed VSI OpenVMS ServiceControl on their systems. Read this manual before you install, upgrade or use version V3.5 of VSI OpenVMS ServiceControl.

Document Structure

- Chapter 1 contains installation information
- Chapter 2 contains support information
- Chapter 3 New Features
- Chapter 4 Bug Fixes

Related Documents

The manual listed below is available in support of this version:

- VSI OpenVMS ServiceControl V3.5 – Users's Guide

This manual is in *Portable Document Format* (PDF) and is included in the OSC034.A installation kit.

Conventions Used in this Manual

Special	in examples indicates text that the system displays or user type input.
UPCASE	in a command represents text that you have to enter as shown.
<i>Lowercase Italics</i>	indicates variable information that a user supplies.
[]	in a command definition, enclose parts of the command that a user can omit.
Key	indicates a named key on the keyboard; for example, RETURN
CTRL/x	is the symbol used to represent the pressing of a control key. It indicates that the user holds down the key marked Ctrl and simultaneously pressing the appropriate key.

Installation Release Notes

This chapter contains information you need to know before installing or upgrading to VSI OpenVMS ServiceControl V3.5.

VSI recommends that you read the following manual before installing or upgrading to VSI OpenVMS ServiceControl V3.5:

- VSI OpenVMS ServiceControl V3.5 – Release Notes (this manual)
- VSI OpenVMS ServiceControl V3.5 – Users's Guide

1.1 Upgrade Path

You can directly upgrade to VSI OpenVMS ServiceControl V3.5 from any previous VSI OpenVMS ServiceControl release.

1.2 Installing VSI OpenVMS ServiceControl

Make sure that you are logged into the system as a privileged user. VSI recommends that you are logged into the SYSTEM account. Install VSI OpenVMS ServiceControl using the VMSINSTAL utility:

```
$ @SYS$UPDATE:VMSINSTAL OSC034 disk:[kit-directory]
```

The installation procedure prompts you only for the disk where VSI OpenVMS ServiceControl will be installed.

It is strongly recommended to install VSI OpenVMS on a cluster common disk. Otherwise VSI OpenVMS ServiceControl has to be installed on each cluster member separately. In addition, if you do not install VSI OpenVMS ServiceControl on a cluster common disk you are responsible for distributing changes of the configuration database within the cluster by copying the configuration database file into the OSC\$CFG directories on each cluster member.

1.3 *Upgrading VSI OpenVMS ServiceControl*

Make sure that you are logged into the system as a privileged user. VSI recommends that you are logged into the SYSTEM account. Upgrade to VSI OpenVMS ServiceControl V3.5 using the VMSINSTAL utility:

```
$ @SYS$UPDATE:VMSINSTAL OSC034 disk:[kit-directory]
```

The upgrade procedure does not prompt the user for any input.

Important

Make sure that OSC has been shutdown cluster-wide before you upgrade to OSC V3.5. Otherwise the upgrade procedure will fail.

1.4 *Upgrade post installation tasks*

1.4.1 **Convert existing OSC configuration databases**

If you have upgraded VSI OpenVMS ServiceControl to V3.5 all OSC configuration databases have to be converted to the new format required by VSI OpenVMS ServiceControl V3.5 before starting OSC again.

Important

OSC V3.5 will fail to start if the default OSC configuration database has not been converted to the new database format.

To convert all existing (default and working) OSC configuration databases start the OSC configuration utility:

```
$ RUN OSC$BIN:OSC$CFG
```

During the initialization phase of the OSC\$CFG utility all OSC configuration databases are converted to the new format. No additional manual actions are required.

1.4.2 **Increase SYSTEM account quota settings**

VSI recommends the following minimum quota settings (or higher) for the SYSTEM account on all OSC cluster members:

Quota	Value
WSdef	8192
WSQuo	32768
WSExtent	65536
Pgflquo	1500000

For more detailed background information regarding the reasons to increase these SYSTEM account quotas please refer to section 3.5.

Supported Versions

VSI OpenVMS ServiceControl V3.5 is supported on:

- HP OpenVMS V7.3-2 AXP with TCP/IP V5.4 ECO5 and scalable kernel enabled
- HP OpenVMS V8.2 AXP
- HP OpenVMS V8.3 AXP
- HP OpenVMS V8.4 AXP
- HP OpenVMS V8.2 IA64
- HP OpenVMS V8.2-1 IA64
- HP OpenVMS V8.3 IA64
- HP OpenVMS V8.3-1H1 IA64
- HP OpenVMS V8.4 IA64
- VSI OpenVMS V8.4-1H1 IA64
- VSI OpenVMS V8.4-2 IA64
- VSI OpenVMS V8.4-2L1 IA64

2.1 Advisory for running VSI OpenVMS ServiceControl on OpenVMS V7.3-2

If one OpenVMS cluster member is running OpenVMS V7.3-2 with TCP/IP V5.4 please ensure that at least ECO5 of TCP/IP V5.4 is installed and that the TCP/IP scalable kernel is enabled.

Important

Do not install and run VSI OpenVMS ServiceControl on a node if either the traditional kernel of TCP/IP V5.4 is enabled or the ECO level is below ECO5.

Due to a bug in TCP/IP for OpenVMS V5.4, that has been fixed with ECO5 in the scalable kernel, but that has never been fixed in the traditional kernel, any attempt to establish a console connection via the OSC\$MGR utility or the OSC management GUI to OSC master control process will fail.

To enable the OpenVMS TCP/IP V5.4 scalable kernel add the following lines to the SYS\$MANAGER:SYSLOGICALS.COM command procedure.

```
$ ! ONLY the argument "PERF=ALL" is supported.  
$ ! Other values may cause unpredictable results  
$ ! to disable scalable kernel support, comment out next line and reboot.  
$ DEFINE/SYSTEM/EXECUTIVE TCPIP$STARTUP_CPU_IMAGES "PERF=ALL"
```

If the TCP/IP Services have already been started when this change is made, a reboot of the system is required for the parameter settings to take effect.

2.2 OSC Management GUI advisory

One of the key new features in this release is that now the user has full control of all transactions currently in progress on the OSC cluster. The new SHOW TRANSACTION command provides an overview of all ongoing transactions and the CANCEL TRANSACTION can be utilized to cancel either a specific or all transactions in progress.

If a transaction is canceled the state of all resources currently executing an action routine (i.e. ONLINE) and their entire parent Services and Service Groups are marked as CANCELED and ADMIN_WAIT.

This new CANCELED state is unknown to the OSC Management GUI V2.1.1 or lower versions. Hence, it is strongly recommended that the OSC management GUI is upgraded to V2.2.0 or higher.

New Features

This chapter contains information about new features in VSI OpenVMS ServiceControl V3.5.

3.1 *OSC agents*

3.1.1 **New resource attribute `ScriptExecUser` – User to run DCL script action routines can be defined resource specific**

OSC agents monitor resources of a particular type, carry out specific operations on resources on behalf of the OSC service engine (online, offline a resource) and cleanup resources in response to resource faults. The OSC agent framework provides the processing logic. The resource type specific logic to monitor, online, offline and cleanup a resource is provided by resource type specific action routines called by the OSC agents. These resource type specific action routines can be either compiled into the agent itself or can be implemented as DCL scripts.

If, prior to OpenVMS ServiceControl V3.5, a DCL script was configured for an action routine of a particular resource, this DCL script was solely executed in the user context of the OSC agent.

The disadvantage of this implementation is, that if different resources are to be started using the same DCL script (online action routine), but it is also a requirement that for each of these resources this start DCL script is executed in a different user context, then different OSC agents have to be created for each of the resources to meet this requirement.

Starting with V3.5 a new resource attribute called **`ScriptExecUser`** is available. Now the user account used to execute DCL action scripts can be defined at the resource level. If this attribute is defined for a particular resource the OSC agent executes the DCL action scripts in the context of the user defined in this attribute. If the attribute value defined is invalid (user defined is not found in SYSUAF), or this attribute is empty, the user context of the OSC agent is used (this is equivalent to the pre-V3.5 behavior) to run the DCL action scripts.

Note

This attribute has no effect on compiled action routines. Compiled action routines are started in the same user context as the OSC agent has been started with.

3.2 *OSC management utility (OSC\$MGR)*

3.2.1 Parameter list and wildcard support

Prior to OpenVMS ServiceControl V3.5 only dedicated single Service Groups, Services or Resources could be applied to the following commands.

- ONLINE SRVGRP <parameter>
- ONLINE SERVICE <parameter>
- ONLINE RESOURCE <parameter>
- OFFLINE SRVGRP <parameter>
- OFFLINE SERVICE <parameter>
- OFFLINE RESOURCE <parameter>
- CLEAR SRVGRP <parameter>
- CLEAR SERVICE <parameter>
- CLEAR RESOURCE <parameter>
- FREEZE SRVGRP <parameter>
- FREEZE SERVICE <parameter>
- FREEZE RESOURCE <parameter>
- UNFREEZE SRVGRP <parameter>
- UNFREEZE SERVICE <parameter>
- UNFREEZE RESOURCE <parameter>
- ENABLE SRVGRP <parameter>
- ENABLE SERVICE <parameter>
- ENABLE RESOURCE. <parameter>
- DISABLE SRVGRP <parameter>
- DISABLE SERVICE <parameter>
- DISABLE RESOURCE <parameter>

These commands did neither accept a parameter list, nor did they support wildcard characters in the parameter string. E.g. the ONLINE SRVGRP command was rejected if the following parameters were entered:

```
OSC$MGR> ONLINE SRVGRP BURG%, E*, VSI$GATEWAY
%CLI-W-NOLIST, list of parameter values not allowed - check use of comma (,)
```

With OpenVMS ServiceControl V3.5 these commands accept comma-separated parameter lists, and full wildcard support is provided for each element of the parameter input list. Asterisk (*) and percent sign (%) wildcard characters can be placed anywhere within each item of the comma-separated parameter list.

Thus, in V3.5 the ONLINE SRVGRP command shown above will start all Service Groups named either BURG%, E* or VSI\$GATEWAY.

3.2.2 New SHOW TRANSACTION command

One of the key new features in this release is that now, the user has full control of all transactions currently in progress in the OSC cluster.

The SHOW TRANSACTION command provides an overview of all ongoing transactions in the OSC cluster. This command displays the transaction state (GOING ONLINE, GOING OFFLINE etc.), the transaction ID, the name of the OSC entity (Service Group, Service or Resource) the transaction is acting upon, the start time of the transaction and whether the transaction was started by a user or by the OSC master control.

If no qualifier or the /ALL qualifier is applied, all transactions are displayed ordered by Service Group, Service and Resources.

If the /SRVGRP qualifier is applied only active Service Group transactions are displayed.

If the /SERVICE qualifier is applied only active Service transactions are displayed.

If the /RESOURCE qualifier is applied only active resource transactions are displayed.

Examples:

```
OSC$MGR> SHOW TRANSACTION
```

TxID	SrvGrp	Transaction	Start Time	Type
1724	ES	GOING ONLINE	22-JAN-2015 12:09:29	User
1725	TX	GOING OFFLINE	22-JAN-2015 12:10:01	Auto
1729	DK	GOING OFFLINE	22-JAN-2015 12:11:59	User

TxID	Service	Transaction	Start Time	Type
1723	HR_KL	GOING ONLINE	22-JAN-2015 12:09:19	User
1726	IL_KL	GOING OFFLINE	22-JAN-2015 12:10:18	User

TxID	Resource	Transaction	Start Time	Type
1727	PRC::EW_KL_ALLO	GOING ONLINE	22-JAN-2015 12:10:50	User
1728	PRC::NW_KL_DB_READ	GOING OFFLINE	22-JAN-2015 12:11:32	User

```
OSC$MGR> SHOW TRANSACTION /SRVGRP
```

TxID	SrvGrp	Transaction	Start Time	Type
1724	ES	GOING ONLINE	22-JAN-2015 12:09:29	User
1725	TX	GOING OFFLINE	22-JAN-2015 12:10:01	Auto
1729	DK	GOING OFFLINE	22-JAN-2015 12:11:59	User

```
OSC$MGR> SHOW TRANSACTION /SERVICE
```

TxID	Service	Transaction	Start Time	Type
1723	HR_KL	GOING ONLINE	22-JAN-2015 12:09:19	User
1726	IL_KL	GOING OFFLINE	22-JAN-2015 12:10:18	User

```
OSC$MGR> SHOW TRANSACTION /RESOURCE
```

TxID	Resource	Transaction	Start Time	Type
1727	PRC::EW_KL_ALLO	GOING ONLINE	22-JAN-2015 12:10:50	User
1728	PRC::NW_KL_DB_READ	GOING OFFLINE	22-JAN-2015 12:11:32	User

3.2.3 New CANCEL TRANSACTION command

The CANCEL TRANSACTION command cancels either a single or all transactions currently in progress.

If the /ID qualifier is applied the transaction associated with the transaction ID assigned to the qualifier is canceled. To display the transaction IDs of all ongoing transactions execute the SHOW TRANSACTION command.

If the /ALL qualifier is applied all transactions are canceled OSC cluster-wide.

If the /NODE qualifier in combination with /ALL qualifier is applied all transactions currently executing on the specified node are canceled.

The /ID and /NODE qualifiers, as well the combination of /ID and /ALL qualifiers are mutually exclusive.

If neither the /ID nor the /ALL qualifier is applied the command fails.

If one starts a transaction in synchronous mode¹ and this transaction is canceled, SS\$_CANCEL is returned to the caller of this transaction signaling, that the transaction has been canceled by another user.

¹ An OSC command is started in synchronous mode if the command is called from the DCL command line prompt or if the DEFINE MODE/SYNC was executed prior to the command. For more information about the DEFINE MODE command please refer to the OSC\$MGR command line help.

If a transaction is canceled all associated resource action routines currently executing (except the monitor action routines) are immediately stopped by OSC. These resources and their respective parent Services and Service Groups are marked as CANCELED. Since the action routines are stopped immediately by OSC these routines have not responded with a valid exit code and so from the OSC perspective these resources are deemed to be in a limbo state. Thus, also the ADMIN_WAIT bit is set in the state fields of these resources and their respective parent Services and Service Groups.

Note

Thus, if a resource is marked as CANCELED and ADMIN_WAIT, verify manually whether or not the system resource(s) referenced by the OSC resource is in a valid state before the ADMIN_WAIT state is cleared.

Only privileged OSC users can execute the CANCEL TRANSACTION command.

Example1: Canceling a single transaction

```
OSC$MGR> SHOW TRANSACTION
TxID   SrvGrp      Transaction      Start Time      Type
-----
8415   EC          GOING ONLINE    22-JAN-2015 14:10:22  User
8416   EM          GOING ONLINE    22-JAN-2015 14:10:22  User
8417   ES          GOING ONLINE    22-JAN-2015 14:10:22  User
8418   EW          GOING ONLINE    22-JAN-2015 14:10:22  User
8419   EZ          GOING ONLINE    22-JAN-2015 14:10:22  User
8420   DE          GOING ONLINE    22-JAN-2015 14:10:22  User
8421   DK          GOING ONLINE    22-JAN-2015 14:10:22  User
8422   DX          GOING ONLINE    22-JAN-2015 14:10:22  User
-----
```

```
OSC$MGR> CANCEL TRANSACTION /ID=8418

...%CONTROL, Received command CANCEL TRANSACTION /ID=8418
...%CONTROL, Transaction CANCEL TRANSACTION queued for execution on VMSTM1
...%CONTROL, Transaction CANCEL TRANSACTION queued for execution on VMSTM2
...%CONTROL, Transaction CANCEL TRANSACTION queued for execution on VMSTM4
...%CONTROL, Transaction CANCEL TRANSACTION starts executing ...
...%CONTROL, Transaction CANCEL TRANSACTION starts executing ...
...%CONTROL, Transaction CANCEL TRANSACTION starts executing ...
...%CONTROL, Cmd: 'CANCEL TRANSACTION/ID=8418 ', Transaction completed
```



```
OSC$MGR> SHOW TRANSACTION
```

TxID	SrvGrp	Transaction	Start Time	Type
8415	EC	GOING ONLINE	22-JAN-2015 14:10:22	User
8416	EM	GOING ONLINE	22-JAN-2015 14:10:22	User
8417	ES	GOING ONLINE	22-JAN-2015 14:10:22	User
8419	EZ	GOING ONLINE	22-JAN-2015 14:10:22	User
8420	DE	GOING ONLINE	22-JAN-2015 14:10:22	User
8421	DK	GOING ONLINE	22-JAN-2015 14:10:22	User
8422	DX	GOING ONLINE	22-JAN-2015 14:10:22	User

```
OSC$MGR>
```

Example2: Canceling all ongoing OSC cluster-wide transactions

```
OSC$MGR> SHOW TRANSACTION
```

TxID	SrvGrp	Transaction	Start Time	Type
8415	EC	GOING ONLINE	22-JAN-2015 14:10:22	User
8416	EM	GOING ONLINE	22-JAN-2015 14:10:22	User
8417	ES	GOING ONLINE	22-JAN-2015 14:10:22	User
8418	EW	GOING ONLINE	22-JAN-2015 14:10:22	User
8419	EZ	GOING ONLINE	22-JAN-2015 14:10:22	User
8420	DE	GOING ONLINE	22-JAN-2015 14:10:22	User
8421	DK	GOING ONLINE	22-JAN-2015 14:10:22	User
8422	DX	GOING ONLINE	22-JAN-2015 14:10:22	User

```
OSC$MGR> CANCEL TRANSACTION/ALL
```

```
...%CONTROL, Received command CANCEL TRANSACTION/ALL
...%CONTROL, Transaction CANCEL TRANSACTION queued for execution on VMSTM1
...%CONTROL, Transaction CANCEL TRANSACTION queued for execution on VMSTM2
...%CONTROL, Transaction CANCEL TRANSACTION queued for execution on VMSTM4
...%CONTROL, Transaction CANCEL TRANSACTION starts executing ...
...%CONTROL, Transaction CANCEL TRANSACTION starts executing ...
...%CONTROL, Transaction CANCEL TRANSACTION starts executing ...
...%CONTROL, Cmd: 'CANCEL TRANSACTION ', Transaction completed
```

```
OSC$MGR> SHOW TRANSACTION
```

```
OSC$MGR>
```

Example3: Canceling all ongoing OSC on node VMSTM1

```
OSC$MGR> SHOW TRANSACTION
```

TxID	SrvGrp	Transaction	Start Time	Type
8415	EC	GOING ONLINE	22-JAN-2015 14:10:22	User
8416	EM	GOING ONLINE	22-JAN-2015 14:10:22	User
8417	ES	GOING ONLINE	22-JAN-2015 14:10:22	User
8418	EW	GOING ONLINE	22-JAN-2015 14:10:22	User
8419	EZ	GOING ONLINE	22-JAN-2015 14:10:22	User
8420	DE	GOING ONLINE	22-JAN-2015 14:10:22	User
8421	DK	GOING ONLINE	22-JAN-2015 14:10:22	User

```
8422   DX                               GOING ONLINE      22-JAN-2015 14:10:22   User
-----
```

```
OSC$MGR> CANCEL TRANSACTION/ALL/node=VMSTM1

...%CONTROL, Received command CANCEL TRANSACTION/ALL
...%CONTROL, Transaction CANCEL TRANSACTION queued for execution on VMSTM1
...%CONTROL, Transaction CANCEL TRANSACTION starts executing ...
...%CONTROL, Cmd: 'CANCEL TRANSACTION ', Transaction completed
```

```
OSC$MGR> SHOW TRANSACTION
```

```
TxID   SrvGrp      Transaction      Start Time      Type
-----
8415   EC          GOING ONLINE    22-JAN-2015 14:10:22   User
8416   EM          GOING ONLINE    22-JAN-2015 14:10:22   User
8417   ES          GOING ONLINE    22-JAN-2015 14:10:22   User
-----
```

```
OSC$MGR>
```

In this example some transactions are still active after the cancel command has been executed since these transactions are executing on other cluster-members.

Please note that once a transaction has been cancelled it is no longer visible via the OSC\$MGR or the GUI interface. A history of the cancelled transaction can be found in the OSC event log files OSC\$LOG:OSC\$EVENTS.LOG_<date>.

3.2.4 New Category column in SHOW SRVGRP output

The output of the SHOW SRVGRP command displays the service group category as shown below:

```
OSC$MGR> SHOW SRVGRP

SrvGrp          Category      Summary State
-----
VSI$MGT         Failover     OFFLINE | TOO FEW INSTANCES
VSI$OLP-COM#1   Failover     ONLINE
VSI$OLP-COM#4   Failover     ONLINE
VSI$OLP-EXEC    Parallel     ONLINE
```

3.2.5 New Critical column in SHOW RESOURCE output

The output of the SHOW RESOURCE command displays whether or not the resources are critical as shown below:

OSC\$MGR> SHOW RESOURCE

Node	Resource	Crit.	State
VMSTM1	FAILIP::VSI\$OLP\$GTY_11	Yes	ONLINE
	FAILIP::VSI\$OLP\$GTY_12	Yes	ONLINE
	FAILIP::VSI\$OLP\$GTY_41	Yes	OFFLINE
	FAILIP::VSI\$OLP\$GTY_42	Yes	OFFLINE
	FSYS::OGS\$CFG	Yes	ONLINE
	FSYS::OGS\$TRACE	Yes	ONLINE
	OGS::VSI\$MGT\$ASTMGR	Yes	OFFLINE
	OGS::VSI\$MGT\$GTY	Yes	OFFLINE

3.3 OSC configuration utility (OSC\$CFG)

3.3.1 Attribute data field extension

In OpenVMS ServiceControl V3.5 the data field size of the attributes listed below have been extended. For a detailed description of these attributes please refer to the VSI OpenVMS ServiceControl V3.5 – Users's Guide manual section A7, A8 and A9:

Resource attributes:

Attribute	Old Size	New Size
ServiceMember	256	1024
CluLckResDisAllow	256	1024

Service attributes:

Attribute	Old Size	New Size
ServiceDependency	256	1024

Service Group attributes:

Attribute	Old Size	New Size
SrvGrpNodes	256	1024
SrvGrpDisAllow	256	1024

Note

Due to these attribute size extensions, the existing OSC configuration databases must be converted after VSI OpenVMS ServiceControl has been upgraded to V3.5. Please refer to section 1.4.1 of how to convert existing OSC configuration databases.

3.3.2 OSC configuration databases are automatically converted to new format.

If, prior to V3.5, OSC was upgraded to a new version all OSC configuration databases had to be manually converted to the new format required by the new OSC release.

Now all OSC configuration databases are automatically converted the first time the OSC\$CFG utility is started after the upgrade to V3.5. During the

initialization phase of the OSC\$CFG utility all OSC configuration databases are checked whether or not they match the new format and are converted if required.

3.3.3 New /ATTRIBUTE qualifier for the MODIFY command

For the MODIFY commands listed below the qualifier /ATTRIBUTE can be applied:

- MODIFY SRVGRP <parameter>
- MODIFY SERVICE <parameter>
- MODIFY RESOURCE <parameter>

The /ATTRIBUTE qualifier is a list qualifier that can be applied to modify several attributes at once for all OSC items addressed in the command parameter. To address more than one attribute enter the attribute/value pairs as a comma separated list enclosed with parenthesis.

If the /ATTRIBUTE qualifier is applied the commands listed above accept comma-separated parameter lists, and full wildcard support is provided for each element of the parameter input list.

Example:

In this example the resource attributes **ScriptExecUser** and **Critical** for all PRC::D%_KL_ALLO, PRC::E*_ALLO and PRC::RU_ISC resources is modified.

```
OSC$CFG> MODIFY RESOURCE -
_OSC$CFG> PRC::D%_KL_ALLO, PRC::E*_ALLO, PRC::RU_ISC -
_OSC$CFG> /ATTRIBUTE=(ScriptExecUser="VSI$EXEC",Critical=TRUE)
```

```
OSC$CFG-I-UPDATTR, successfully updated resource PRC::DE_KL_ALLO
OSC$CFG-I-UPDATTR, successfully updated resource PRC::DK_KL_ALLO
OSC$CFG-I-UPDATTR, successfully updated resource PRC::DX_KL_ALLO
OSC$CFG-I-UPDATTR, successfully updated resource PRC::EC_KL_ALLO
OSC$CFG-I-UPDATTR, successfully updated resource PRC::EM_KL_ALLO
OSC$CFG-I-UPDATTR, successfully updated resource PRC::ES_KL_ALLO
OSC$CFG-I-UPDATTR, successfully updated resource PRC::EW_KL_ALLO
OSC$CFG-I-UPDATTR, successfully updated resource PRC::EZ_KL_ALLO
OSC$CFG-I-UPDATTR, successfully updated resource PRC::RU_ISC
```

3.3.4 New COPY SRVGRP command

Format:

```
COPY SRVGRP srvgrp-name new-srvgrp
```

The COPY SRVGRP command copies an existing service group defined by the *srvgrp-name* parameter to a new service group defined by the *new-srvgrp* parameter.

The use of wildcards is not permitted.

Example:

```
OSC$CFG> COPY SRVGRP VSI$OLP-EXEC VSI$OLP-EXEC_SRVGRP
OSC$CFG-I-COPY, successfully copied SrvGrp 'HP$OLP-EXEC' to
'VSI$OLP-EXEC_SRVGRP'
```

3.3.5 New COPY SERVICE command

Format:

```
COPY SERVICE service-name new-service
```

The COPY SERVICE command copies an existing service defined by the *service-name* parameter to a new service defined by the *new-service* parameter.

The use of wildcards is not permitted.

Example:

```
OSC$CFG> COPY SERVICE VSI$OLP-EXEC VSI$OLP-EXEC_SER
OSC$CFG-I-COPY, successfully copied Service 'VSI$OLP-EXEC' to
'VSI$OLP-EXEC_SER'
```

3.3.6 New COPY RESOURCE command

Format:

```
COPY RESOURCE resource-name new-resource
```

The COPY RESOURCE command copies an existing resource defined by the *resource-name* parameter to a new resource defined by the *new-resource* parameter.

The resource type cannot be modified with this command. Thus, if the resource type in the *resource-name* and the *new-resource* parameter differ the command fails.

If node specific definitions exist of the resource to be copied and the *resource-name* and *new-resource* parameter does not address a specific node, all existing node specific resource definitions are copied (see example below).

If the *old-name* parameter addresses a node specific resource but the *new-name* parameter does not, or vice versa, the command fails.

For detailed information about the OSC resource naming rules please refer to the online help of the ADD RESOURCE command.

The use of wildcards is not permitted.

Example:

In this example the resource FAILIP::VSI\$OLP\$GTY_11 and all its node specific definitions are copied to FAILIP::VSI\$OLP\$GTY_13:

```
OSC$CFG> COPY RESOURCE FAILIP::VSI$OLP$GTY_11 -
_OSC$CFG> FAILIP::VSI$OLP$GTY_13
OSC$CFG-I-COPY, successfully copied Resource
'FAILIP::VSI$OLP$GTY_11@VMSTM1' to
'FAILIP::VSI$OLP$GTY_13@VMSTM1'
OSC$CFG-I-COPY, successfully copied Resource
'FAILIP::VSI$OLP$GTY_11@VMSTM4' to
'FAILIP::VSI$OLP$GTY_13@VMSTM4'
```

In the next example, the VMSTM1 node specific definition for the resource FAILIP::VSI\$OLP\$GTY_11 is copied to the VMSTM2 node specific resource FAILIP::VSI\$OLP\$GTY_13.

```
OSC$CFG> COPY RESOURCE FAILIP::VSI$OLP$GTY_11@VMSTM1 -
_OSC$CFG> FAILIP::VSI$OLP$GTY_13@VMSTM2
OSC$CFG-I-COPY, successfully copied Resource
'FAILIP::VSI$OLP$GTY_11@VMSTM1' to
'FAILIP::VSI$OLP$GTY_13@VMSTM2'
```

3.3.7 New RENAME SRVGRP command

Format:

```
RENAME SRVGRP old-name new-name
```

The RENAME SRVGRP command renames an existing service group defined by the *old-name* parameter to the name defined by the *new-name* parameter.

In addition the service group membership of all services which are members of the *old-name* service group is updated with the *new-name* service group (see example below).

The use of wildcards is not permitted.

Example:

```
OSC$CFG> RENAME SRVGRP VSI$OLP-EXEC VSI$OLP-EXEC_SRVGRP
OSC$CFG-I-RENAME, successfully renamed SrvGrp 'VSI$OLP-EXEC' to
                'VSI$OLP-EXEC_SRVGRP'
OSC$CFG-I-RENAME, Service 'VSI$OLP-EXEC', replaced 'VSI$OLP-EXEC'
with
                'VSI$OLP-EXEC_SRVGRP' in SrvGrp membership list
```

3.3.8 New RENAME SERVICE command

Format:

```
RENAME SERVICE old-name new-name
```

The RENAME SERVICE command renames an existing service defined by the *old-name* parameter to the name defined by the *new-name* parameter.

In addition the service membership of all resources which are members of the *old-name* service is updated with the *new-name* service. If the renamed service is a child service of other services the service dependency list of its parent services is also automatically updated (see example below).

The use of wildcards is not permitted.

Example:

```
OSC$CFG> RENAME SERVICE BE_BASE BE_BASE_SER
OSC$CFG-I-RENAME, successfully renamed Service 'BE_BASE' to
'BE_BASE_SER'
OSC$CFG-I-RENAME, Service 'BE_IFMS', replaced 'BE_BASE' with
                'BE_BASE_SER' in Service dependency list
OSC$CFG-I-RENAME, Resource 'FAILIP::BE_IP@H1HA07', replaced 'BE_BASE'
with 'BE_BASE_SER' in Service membership list
OSC$CFG-I-RENAME, Resource 'FAILIP::BE_IP@H1HA08', replaced 'BE_BASE'
with 'BE_BASE_SER' in Service membership list
OSC$CFG-I-RENAME, Resource 'PRC::FAILSAFEIP', replaced 'BE_BASE' with
                'BE_BASE_SER' in Service membership list
OSC$CFG-I-RENAME, Resource 'QUEUE::BE_BATCH', replaced 'BE_BASE' with
                'BE_BASE_SER' in Service membership list
OSC$CFG-I-RENAME, Resource 'RLIST::BE_INSTANCE', replaced 'BE_BASE'
with 'BE_BASE_SER' in Service membership list
```



```
OSC$CFG-I-RENAME, Resource 'SHD::BEDISK', replaced 'BE_BASE' with
'BE_BASE_SER' in Service membership list
```

3.3.9 New RENAME RESOURCE command

Format:

```
RENAME RESOURCE old-name new-name
```

The RENAME RESOURCE command renames an existing resource defined by the *old-name* parameter to the name defined by the *new-name* parameter.

If the resource is a child resource of other resources the resource dependency list of its parent resources is automatically updated (see example below).

The resource type cannot be modified with this command. Thus, if the resource type in the *old-name* and the *new-name* parameter differ the command fails.

If node specific definitions exist of the resource to be renamed and the *old-name and new-name* parameter does not address a specific node, all existing node specific resource definitions are renamed (see example below).

If the *old-name* parameter addresses a node specific resource but the *new-name* parameter does not, or vice versa, the command fails.

For detailed information about the OSC resource naming rules please refer to the online help of the ADD RESOURCE command.

The use of wildcards is not permitted.

Example:

In this example the resource FAILIP::VSI\$OLP\$GTY_11 and all its node specific definitions are renamed to FAILIP::VSI\$OLP\$GTY_13:

```
OSC$CFG> RENAME RESOURCE FAILIP::VSI$OLP$GTY_11
FAILIP::VSI$OLP$GTY_13
OSC$CFG-I-RENAME, successfully renamed Resource
'FAILIP::VSI$OLP$GTY_11@VMSTM1' to
'FAILIP::VSI$OLP$GTY_13@VMSTM1'
OSC$CFG-I-RENAME, successfully renamed Resource
'FAILIP::VSI$OLP$GTY_11@VMSTM4' to
'FAILIP::VSI$OLP$GTY_13@VMSTM4'
OSC$CFG-I-RENAME, Resource 'OGS::VSI$OLP$GTY_11', replaced
'FAILIP::VSI$OLP$GTY_11' with
'FAILIP::VSI$OLP$GTY_13'
```

in resource dependency list

In the next example the VMSTM1 node specific definition of the resource FAILIP::VSI\$OLP\$GTY_11 is renamed to FAILIP::VSI\$OLP\$GTY_13 for node VMSTM2.

```
OSC$CFG> RENAME RESOURCE FAILIP::VSI$OLP$GTY_11@VMSTM1 -
_OSC$CFG> FAILIP::VSI$OLP$GTY_13@VMSTM2
OSC$CFG-I-RENAME, successfully renamed Resource
                'FAILIP::VSI$OLP$GTY_11@VMSTM1' to
                'FAILIP::VSI$OLP$GTY_13@VMSTM2'
OSC$CFG-I-RENAME, Resource 'OGS::VSI$OLP$GTY_11', replaced
                'FAILIP::VSI$OLP$GTY_11' with
                'FAILIP::VSI$OLP$GTY_13'
                in resource dependency list
```

3.4 *OSC GUI connectivity server*

The OSC GUI Connectivity server (OSC\$ODBC TCP/IP service) is the interface between the OSC master control and the VSI OpenVMS Service Control Management GUI.

When the GUI connects to the OSC GUI Connectivity server it requests topology and state information for all OSC configuration items. Prior to V3.5 the OSC GUI Connectivity server forwarded the response of the OSC master control synchronously to the GUI. OSC topology and state information requests block OSC master control activities until the OSC master control process has sent all data to the requestor. If the OSC master control is blocked neither OSC state changes nor user requests are processed. Thus, if prior to VSI OpenVMS ServiceControl V3.5 the network latency between the GUI and the OSC connectivity server was high, the OSC master control could be blocked for several seconds up to several minutes when a GUI connected to an OSC cluster.

In V3.5 reading data from the OSC master control and transmitting the data to the GUI has been decoupled in the OSC GUI Connectivity server by buffering the data received from the OSC master control. Now, the OSC master control is typically blocked for less than one second whenever a GUI connects to the OSC cluster.

The OSC GUI Connectivity server is started under the SYSTEM account. To receive the maximum value out of this new feature, please ensure that the memory and page file settings of the SYSTEM account are sufficient so that the OSC GUI Connectivity server can buffer all data records received in response to an OSC topology and state information request in memory. The memory and page file settings required depend on the number of resources and the dependency depth of the Services and Service Groups configured.

One can test whether or not the memory and page file quota settings of the SYSTEM account are sufficient for the current OSC configuration:

1. Start the OSC GUI
2. Connect to the OSC cluster
3. Wait until all Service Groups, Services and Resources are displayed in the GUI
4. Disconnect the GUI
5. Search for the string "OSC\$ODBC-W-INSFMEM" in the log file of the OSC GUI Connectivity server on the OpenVMS server that you were connected to.

Log file: [OSC\\$LOG:OSC\\$ODBC_<node-name>.LOG](#)

6. If no matches are found the quota settings are sufficient, otherwise the quotas – especially the page file quota has to be increased.

However, this analysis does not guarantee, that the quota settings will still be sufficient once the OSC configuration changes. Thus, it is best practice to increase the quotas of the SYSTEM account on all OSC cluster members to the values recommended by VSI after VSI OpenVMS ServiceControl V3.5 has been installed.

VSI recommends the following minimum SYSTEM account quota settings (or higher):

Quota	Value
WSdef	8192
WSQuo	32768
WSExtent	65536
Pgflquo	1500000

3.5 OSC event notification

3.5.1 OSC entity type passed in P8 to DCL user event scripts

With VSI OpenVMS ServiceControl V3.5 eight parameters are passed to DCL user event scripts. P8 contains the OSC entity type in a state change event. Possible keywords passed in P8:

- RESOURCE (resource state change)
- SERVICE (service state change)
- SRVGRP (service group state change)

Please note that only the OSC state change event classes listed below pass either of these keywords to DCL user event scripts:

- OSCAGT_STATE_EVT
- OSCSRV_STATE_EVT
- OSCCTRL_STATE_EVT

All other event classes provide a zero length string in P8.

Bug Fixes

This chapter contains information about bug fixes available in VSI OpenVMS ServiceControl V3.5

4.1 OSC master control engine (OSC\$CTRL)

4.1.1 Transaction does not terminate if an OSC service engine is restarted or fails while executing this transaction

If, prior to OpenVMS Service Control V3.5, an OSC service engine was restarted or failed for any reason while a transaction was processed by this OSC service engine, this transaction was not removed from the OSC master control transaction queue. Thus, such transactions did not complete until the OSC master control engine was restarted.

Now all transactions terminate immediately and return SS\$_OPINCOMPL to the caller signaling that the transaction has failed, if an OSC service engine is restarted or fails while it processes these transactions.

4.1.2 Critical resource of a failover group fails during initial OSC cluster state transition

When OpenVMS Service Control is started cluster-wide the active OSC master control process initiates OSC state transition. During the state transition the active OSC master control process checks if all OSC components are started (OSC service engines, OSC agents) and starts them if required. In addition, the OSC master control process requests the current state of all managed OSC items (resources, services and service groups) from the OSC agents and OSC service engines cluster-wide. If a fault condition of a critical resource of a failover Service Group was reported to the OSC master control, the OSC master control started this Service Group on an alternate OSC cluster member after the initial state transition phase without first trying to stop the Service Group on the node where the resource had failed. Thus, if On-Off resources (except the resource that failed) of such a failover Service Group were already online on the node where a resource failure was detected, these same resources were started on another OSC cluster member unconditionally. This behavior could cause serious system damage and contradicted the intended OSC configuration.

This major bug has been fixed in V3.5. If the OSC master control now detects such a situation, after the initial state transition phase, it first tries to offline a faulted failover or multi-instance Service Group if it detects that at least one On-Off resource of the Service Group is online. Only if

the OFFLINE transaction succeeds, does OSC try to online the Service Group on an alternate host (as defined in the OSC Service Group configuration). If this OFFLINE transaction fails the Service Group is set immediately to the ADMIN_WAIT state.

4.1.3 Starting the same transaction from different OSC consoles

If, prior to OpenVMS Service Control V3.5, an ONLINE or OFFLINE transaction was started on the same Service Group or Service simultaneously from different OSC consoles both transactions were rejected by the OSC master control. Thus, it could have occurred that an ONLINE or OFFLINE transaction was not executed, only because the same transaction was started from two different OSC consoles in parallel and not because the state of the Service Group or Service prevented the execution.

Now if the OSC master control receives the same transaction from different consoles, the first transaction request received will be processed and the second will be rejected (transaction is already in progress).

4.1.4 ONLINE RESOURCE/NODE fails if the resource is a member of a Parallel or MultiInstance Service Group and the Service Group is already online on a node

If, prior to OpenVMS Service Control V3.5, the user wanted to start a resource on a particular node with the ONLINE RESOURCE/NODE command and the resource was a member of a **Parallel** or **MultiInstance** service group which was already online on a node, the command incorrectly failed with:

`%CONTROL, Online state of owner SrvGrp ... inhibits online transaction for resource ...`

4.2 OSC service engine (OSC\$SRV)

4.2.1 If an ONLINE transaction fails the transaction was not removed from the transaction queue

If, prior to OpenVMS Service Control V3.5, an ONLINE transaction failed due to a service or resource failure, the transaction was not removed from the OSC service engine transaction queue and remained in the OSC service transaction queue until the OSC service engine was restarted. Thus, if a subsequent transaction was started on such a resource or service the pending transaction in the OSC service engine transaction queue prevented this new transaction of being processed. This new transaction was rejected signaling that the service or resource was in a busy state.

4.2.2 If an OFFLINE transaction fails the transaction was not removed from the transaction queue

If, prior to OpenVMS Service Control V3.5, an OFFLINE transaction failed due to a service or resource failure, the transaction was not removed from the OSC service engine transaction queue and remained in the OSC service transaction queue until the OSC service engine was restarted. Thus, if a subsequent transaction was started on such a resource or service the pending transaction in the OSC service engine transaction queue prevented this new transaction of being processed. This new transaction was rejected signaling that the service or resource was in a busy state

4.2.3 Invalid resource state reported to OSC master control if an OFFLINE action routine fails

If an OFFLINE command on a resource fails with OFFLINE INEFFECTIVE signaling that the resource is still online, the CLEAN action routine is automatically called by the OSC agent. If the CLEAN action routine succeeds the resource will be set to the OFFLINE state.

Prior to OpenVMS Service Control V3.5, such a resource was marked as faulted by the OSC service engine when the OFFLINE INEFFECTIVE state was reported. This fault state was not cleared if the CLEAN action routine succeeded. This behavior caused Service and Service Groups OFFLINE

transactions to fail even though all of their resources were in an OFFLINE state.

Now, if a resource fails to go offline but the CLEAN action routine succeeds, the resource is no longer marked faulted and its state is correctly set to OFFLINE.

4.2.4 ONLINE transaction fails if a non-critical resource fails to go online

If, prior to OpenVMS Service Control V3.5, an ONLINE transaction was triggered on a Service or Service Group and one of its non-critical resources failed to go online the ONLINE transaction failed.

This behavior is not in line with the basic concept of OpenVMS Service Control. The ONLINE state of a Service or Service Group does not depend on the availability of non-critical resources. Thus, a Service or Service Group ONLINE transaction must not terminate if a non-critical resource fails to go online.

Now, if a non-critical resource fails to go online the OSC service engine sends notification events, but continues with the Service or Service Group ONLINE transaction.

4.2.5 Transaction does not terminate if an OSC agent is restarted or fails while executing an action routine

If, prior to OpenVMS Service Control V3.5, an OSC service engine was processing a transaction and an OSC agent was restarted or failed while an action routine associated with this transaction was processed by the OSC agent, this transaction was not removed from the OSC service engine transaction queue. Thus, such a transaction did not complete until the OSC service engine was restarted.

Now all transactions terminate immediately and return SS\$_OPINCOMPL to the caller signaling that the transaction has failed, if an OSC agent processing transaction related action routines fails or is restarted.

4.4 **OSC agents**

4.4.1 **OnlineRetryLimit is set to 0 if a resource fails to go online**

The **OnlineRetryLimit** resource attribute specifies the number of times the online action routine is retried by an OSC agent if the initial attempt to bring a resource online fails. Each ONLINE retry attempt decrements the value of the attribute.

Prior to OpenVMS ServiceControl V3.5, the value of this attribute was never reset to its initial value. Thus in previous versions, once a resource failed to go online this attribute was decremented and finally set to zero and so the next time an ONLINE transaction on this particular resource was started no online retries were performed. Once the online retry limit was set to zero it remained zero for this resource until the respective OSC agent was restarted.

Now, the **OnlineRetryLimit** resource attribute is reset to its initial configured value at the end of the ONLINE transaction regardless of whether or not the resource failed to go online. Thus, it is guaranteed that whenever an ONLINE transaction is started on a resource the OSC agent always performs the same number of online action retries as defined by the **OnlineRetryLimit** resource attribute.

4.5 *OSC management utility (OSC\$MGR)*

4.5.1 **ONLINE transactions on failover Service Groups return wrong status code**

If, prior to OpenVMS ServiceControl V3.5, the ONLINE SRVGRP command is executed without applying the /NODE qualifier on a failover Service Group with the permission to failover (attribute SrvGrpFailover = TRUE) in synchronous mode¹ and the Service Group failed to go online on the primary node, the command returns SS\$_OPINCOMPL signaling that the transaction has failed regardless if the Service Group has been successfully started on an alternate OSC cluster member.

Now, if the ONLINE SRVGRP command is applied without the /NODE qualifier on such a Service Group, SS\$_NORMAL is returned to the caller if the Service Group has been successfully started on any OSC cluster member. SS\$_OPINCOMPL is only returned if such a Service Group failed to go online on all OSC cluster members.

4.5.2 **ONLINE transactions on parallel Service Groups return to the caller before the Service Groups are online on the all CSC cluster members**

If, prior to OpenVMS ServiceControl V3.5, the ONLINE SRVGRP command is executed without applying the /NODE qualifier on a **Parallel** Service Group in synchronous mode¹, the command returns to the caller immediately when the first OSC cluster member reports, that the ONLINE transaction has either succeeded or failed. Due to this behavior, if SS\$_NORMAL was returned signaling that the operation succeeded, it was not guaranteed that the Service Group had successfully started on all OSC cluster members, but only on one of them.

Now, the ONLINE SRVGRP command executed on a **Parallel** Service Group returns to the caller after the ONLINE transaction has completed on all OSC cluster members. SS\$_NORMAL is returned only if the Service Group is online on all OSC cluster members. If the **Parallel** Service Group fails to go online on at least one of the OSC cluster members, the command returns SS\$_OPINCOMPL.

¹ An OSC command is started in synchronous mode if the command is called from the DCL command line prompt or if the DEFINE MODE/SYNC was executed prior to the command.

4.5.3 **ONLINE transactions on non-critical resources return wrong status code if the transaction fails**

If, prior to OpenVMS ServiceControl V3.5, the ONLINE RESOURCE command is executed on a non-critical resource in synchronous mode¹, the command returns SS\$_NORMAL regardless if the command succeeds or not.

Now, SS\$_NORMAL is returned only if the resource is online when the command completes. Otherwise SS\$_OPINCOMPL is returned signaling that the command failed to start the resource.

4.5.4 **/NODE qualifier is ignored in the ONLINE SERVICE command**

If, prior to OpenVMS Service Control V3.5, the ONLINE SERVICE command is executed and the Service Group the Service belongs to is not online on any OSC cluster member, the command always fails with:

No online SrvGrp for service <service-name> found, command has to be applied with the /NODE qualifier

regardless if the /NODE qualifier was applied or not.

Now, if the /NODE qualifier is applied the service is started on the node defined by the /NODE qualifier provided the Service's ONLINE state will not conflict with the Service Group state the Service is a member of (i.e. some On-Off resources of a failover Service Group are already ONLINE on another OSC cluster member).

4.5.5 **SHOW CLUSTER returns SS\$_INVARG when called from the command line**

Any OSC\$MGR command can be executed from the DCL command line by defining a foreign command symbol. If, prior to OpenVMS Service-Control V3.5, the SHOW CLUSTER was executed in this way from the DCL command line the command failed with:

%SYSTEM-E-INVARG, invalid argument

¹ An OSC command is started in synchronous mode if the command is called from the DCL command line prompt or if the DEFINE MODE/SYNC was executed prior to the command.

Now, the OSC cluster information will be displayed correctly.

4.5.6 SHOW VERSION returns SS\$_INVARG when called from the command line

Any OSC\$MGR command can be executed from the DCL command line by defining a foreign command symbol. If, prior to OpenVMS Service-Control V3.5, the SHOW VERSION was executed in this way from the DCL command line the command failed with:

```
%SYSTEM-E-INVARG, invalid argument
```

Now, the version information will be displayed correctly.

4.5.7 No output displayed when SHOW RESOURCE, SHOW SERVICE or SHOW SRVGRP called from the command line

Any OSC\$MGR command can be executed from the DCL command line by defining a foreign command symbol. If, prior to OpenVMS Service-Control V3.5, one of the SHOW commands listed below were executed to output was displayed on the screen:

- SHOW RESOURCE
- SHOW SERVICE
- SHOW SRVGRP

Now, the resource, service or service group information will be displayed.

4.6 **OSC configuration utility (OSC\$CFG)**

4.6.1 **Loading OSC configuration template definition deletes OSC cluster configuration**

If, prior to OpenVMS ServiceControl V3.5, the OSC configuration template file is applied on an existing OSC configuration using the command:

```
OSC$CFG> LOAD OSC$CFG:OSC$TEMPLATES.CFG
```

the OSC cluster configuration was deleted. Thus, the user had to redefine the OSC cluster configuration.

With V3.5 reloading the OSC configuration template file preserves the existing OSC cluster configuration.

4.6.2 **Votes = 1 is not automatically assigned to all cluster members**

If, prior to OpenVMS ServiceControl V3.5, one defined the OSC cluster members (attribute **OscCtrlNode**) without explicitly assigning votes to each of the OSC cluster members no vote is assigned to the last node in the comma separated list instead of vote 1.

Example:

```
OSC$CFG> MODIFY CLUSTER
```

```
Welcome to the OSC cluster configuration wizard
```

```
-----
```

```
...
```

```
Dscr: OscCtrl node list
```

```
Attr: {OscCtrlNode} []: VMSTM1,VMSTM4
```

Prior to OpenVMS ServiceControl V3.5 vote 1 would have been assigned to VMSTM1 and vote 0 would have been assigned to VMSTM4.

Now both OSC cluster members will have vote 1 assigned.