



## **SAS MOM 6.0 R14 Hot Fix 04 Test Summary Report**

The correct bibliographic citation for this manual is as follows: SAS Institute Inc. 2012. SAS® SolutionName: DocumentTitle.  
Cary, NC: SAS Institute Inc.

SAS® SolutionName: DocumentTitle

Copyright © 2012, SAS Institute Inc., Cary, NC, USA

All rights reserved. Produced in the United States of America.

U.S. Government Restricted Rights Notice: Use, duplication, or disclosure of this software and related documentation by the U.S. government is subject to the Agreement with SAS Institute and the restrictions set forth in FAR 52.227-19, Commercial Computer Software-Restricted Rights (June 1987).

SAS Institute Inc., SAS Campus Drive, Cary, North Carolina 27513.

October 2012

SAS® Publishing provides a complete selection of books and electronic products to help customers use SAS software to its fullest potential. For more information about our e-books, e-learning products, CDs, and hard-copy books, visit the SAS Publishing Web site at [support.sas.com/pubs](http://support.sas.com/pubs) or call 1-800-727-3228.

SAS® and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

Other brand and product names are registered trademarks or trademarks of their respective companies.

**Preface:**

The document mentions the Functional Testing activities performed in the release process of the SAS MOM6.0 R14 HotFix04. This report does not contain any information on Performance, Load & Accessibility testing for SAS MOM 6.0R14 HotFix04.

**Reviewed By:**

- Khalid Karim – Senior Software Manager - Testing, STG

**Created By:**

- SAS-MOM Testing Team, Pune

**Revision History:**

**Created On:** 23<sup>rd</sup> July, 2013  
**Reviewed On:** 23<sup>rd</sup> July, 2013  
**Updated On:** 23<sup>rd</sup> July, 2013

## Table of Contents

|   |    |
|---|----|
| 1. Software Info & Milestone Dates: .....                       | 5  |
| 1.1 Software Classification:.....                               | 5  |
| 1.2 Important Dates:.....                                       | 5  |
| 2. Testing Highlights:.....                                     | 6  |
| 2.1 Defect Included in MOM 6.0R14 HotFix04:.....                | 6  |
| 3. Test Description: .....                                      | 6  |
| 3.1 Manual Test Description: .....                              | 6  |
| 3.2 API Changes tested for Central Folder Deletion Change ..... | 7  |
| 3.3 Installation Test Description: .....                        | 13 |
| 4. Recommendation:.....   | 13 |
| 5. References & Credits: .....                                  | 14 |
| 5.1 References: .....   | 14 |
| 5.2 Credits (MOM Testing Team):.....                            | 14 |

## 1. Software Info & Milestone Dates:

### 1.1 Software Classification:

|                      |   |
|----------------------|---|
| Name:                | SAS Marketing Operations Management (SAS MOM) |
| Major Version no. :  | 6.0   |
| Minor Version no. :  | 14  |
| Version Update no. : | 0   |
| Version HotFix no. : | 04  |

### 1.2. Important Dates:

|                          |                             |
|--------------------------|-----------------------------|
| Test Start Date:         | 14 <sup>th</sup> July, 2013 |
| Testing Completion Date: | 19 <sup>th</sup> July, 2013 |
| Release Date:            | 23 <sup>rd</sup> July, 2013 |

## 2. Testing Highlights:

- Total no. of Iteration Performed : 1
- Total no. of Defects included in 6.0R14HotFix04 : 1
- Total no. of Defects Verified in 6.0R14HotFix04 : 1

### 2.1. Defect Included in MOM 6.0R14 HotFix04:

| No. | Defect ID                | Defect Description                                 | Component      |
|-----|--------------------------|--|----------------|
| 1   | <a href="#">S0991814</a> | CRP:MountPoint for Lilly was deleted on 10July2013 | Infrastructure |

## 3. Test Description:

### 3.1. Manual Test Description:

1. Verification of defects included in 6.0 R14 HotFix04 is done and impacted areas are tested.
2. All the APIs which were changed to go through central folder deletion code were verified to be working as expected, detailed API List is mentioned below.
3. Marketing Workbench is tested using Lilly Configuration.
4. Deletion of Mount Point and XML Mount Point is tried in Unit Testing through central folder deletion code, which is introduced in ALCommonExtension and found to be not able to delete these folders.
5. ExecuteInDoubtTransactions APIs could not be verified since there is no known functional case which traverses this code path.
6. 6.0 R14 HotFix04 is tested using browser as IE9 (32-bit) only.
7. High Level Testing of following modules is covered to ensure that this fix does not have any impact on other functionalities:
  - a. Artwork Producer
  - b. Calendars
  - c. Dashboard
  - d. DAM
  - e. Marketing Workbench
  - f. Strategic Planner
  - g. Approvals
  - h. Reports
  - i. AdHoc Reports
  - j. Resource Management

- k. Claims Management
  - l. Offer Management
  - m. Timesheets
  - n. Resource Planner
  - o. Schema Designer
  - p. UI Framework
  - q. LeftBar Configuration
  - r. Vendor Management
  - s. User Group Management
8. Verified the XML error, Exception log table, Server Event Viewer log.

### 3.2. API Changes tested for Central Folder Deletion Change

| Index | Project         | Location  | API                          | Folder Path in normal cases  | Why Delete Folder is called   | Tested |
|-------|-----------------|---|------------------------------|--|---|--------|
| 1     | ArtworkProducer | Services\Engines\ArtworkProducerEngine\ArtworkProducerEngine\Artwork.cs(2461) | CreateArtwork_HTML           | MediaProcessor\ArtworkCache\<TenantName>\<ArtworkID>\Output\                           | To clean up previous outout.  | Yes    |
| 2     | ArtworkProducer | Services\Engines\ArtworkProducerEngine\ArtworkProducerEngine\Artwork.cs(2500) | CreateArtwork_HTML           | MediaProcessor\ArtworkCache\<TenantName>\<ArtworkID>\Output\Images\                    | To clean up previous outout.  | Yes    |
| 3     | ArtworkProducer | Services\Engines\ArtworkProducerEngine\ArtworkProducerCache.cs(287)           | CreateCacheForFormat         | MediaProcessor\FormatCacheCache\<TenantName>\<FormatID>\Original                       | To clean up previous format file in case of updating with new format file                         | Yes    |
| 4     | ArtworkProducer | Services\Engines\ArtworkProducerEngine\ArtworkProducerCache.cs(464)           | CreateCacheForGraphicElement | MediaProcessor\FormatCacheCache\<TenantName>\<FormatID>\GE_<GraphicElementID>\Original | To clean up previous graphic file in case of updating with new file                               | Yes    |
| 5     | ArtworkProducer | Services\Engines\ArtworkProducerEngine\ArtworkProducerCache.cs(860)           | PrepareArtworkCache          | MediaProcessor\ArtworkCache\<TenantName>\<ArtworkID>                                   | API PrepareArtworkCache is used to re create Artwork Cache, so it clean up the cache if it exist. | Yes    |

|    |                 |  |                                     |   |   |     |
|----|-----------------|--|-------------------------------------|---|---|-----|
| 6  | ArtworkProducer | Services\Engines\ArtworkProducerEngine\ArtworkProducerEngine\ArtworkProducerCache.cs(930)                      | PrepareFormatCache                  | MediaProcessor\FormatCacheCache\<TenantName>\<FormatID>\Original              | API PrepareFormatCache is used to re create Format Cache, so it clean up the cache if it exist.   | Yes |
| 7  | ArtworkProducer | Services\Engines\ArtworkProducerEngine\ArtworkProducerEngine\ArtworkProducerCache.cs(1360)                     | DeleteArtworkCache                  | MediaProcessor\ArtworkCache\<TenantName>\<ArtworkID>                          | In case of DeleteArtwork this API gets called to delete cache related to artwork  | Yes |
| 8  | ArtworkProducer | Services\Engines\ArtworkProducerEngine\ArtworkProducerEngine\ArtworkProducerCache.cs(1391)                     | DeleteFormatCache                   | MediaProcessor\FormatCacheCache\<TenantName>\<FormatID>                       | This API is called from two places. While creating Format if section creation for format fails, it calls delete cache to delete cache. Also in case of delete Format it calls delete cache to remove the cache. | Yes |
| 9  | ArtworkProducer | Services\Engines\ArtworkProducerEngine\ArtworkProducerEngine\ArtworkProducerCache.cs(1428)                     | DeleteGraphicElementFromFormatCache | MediaProcessor\FormatCacheCache\<TenantName>\<FormatID>\GE_<GraphicElementID> | This API is called in case of DeleteGraphic element to clear the cahce related to graphic element   | Yes |
| 10 | ArtworkProducer | Services\QueueComponents\AL_AP_BatchArtworkGenerator\AL_AP_BatchArtworkGenerator\BatchArtworkGenerator.cs(290) | ProcessJob                          | MediaProcessor\BatchArtworkJob\<TenantName>\<JobID>                           | To clean up the processed Job   | Yes |
| 11 | DAM             | Services\Engines\DAMEngine\DAMEngine\Vault.cs(316):  | CreateVault                         | Vault Mount point location  | We call Delete Folder API from Create vault API in case there are error in creating vault (OM Error)  | Yes |

|    |     |  |                    |  |   |     |
|----|-----|--|--------------------|--|---|-----|
| 12 | DAM | Services\Engines\<br>DAMEngine\DAM<br>Engine\Vault.cs(3<br>26):                                | CreateVault        | Vault Mount point<br>location  | We call Delete<br>Folder API from<br>Create vault API<br>in case there is<br>an exception<br>while creating<br>vault and Vault<br>mount point is<br>created .   | Yes |
| 13 | DAM | Services\Engines\<br>DAMEngine\DAM<br>Engine\Vault.cs(5<br>29)                                 | DeleteVault        | Vault Mount point<br>location  | When Vault is<br>deleted from UI,<br>the vault mount<br>point folder<br>should also get<br>deleted. And we<br>make a<br>recursive call to<br>DeleteFolderAPI<br>to delete all the<br><Unique ID><br>folders created<br>inside the vault<br>mount point for<br>catalogged<br>Assets. | Yes |
| 14 | DAM | Services\Engines\<br>DAMEngine\DAM<br>Engine\Vault.cs(8<br>25):                                | UpdateVault        | Vault Mount point<br>location  | We call<br>DeleteFolder<br>API in case there<br>are any errors<br>while updating<br>vault and only if<br>Mount point is<br>updated.   | Yes |
| 15 | DAM | Services\Media<br>Processors\AssetP<br>rocessor\AssetPro<br>cessor\AssetProce<br>ssor.cs(3249) | NTFSJobCleanU<br>p | Catalog Jobpath  | We delete the<br>catalog job after<br>successful<br>processing of<br>job by Cataloger   | Yes |
| 16 | DAM | Services\Media<br>Processors\Check<br>outProcessor\Che<br>ckoutProcessor.C<br>s(1480):         | NTFSJobCleanU<br>p | Input Folder path for ZIP<br>download<br>c:\Temp\plugin\123456<br>77\INPUT | To cleanup the<br>tempaorary<br>location after<br>successful<br>creation of ZIP<br>file   | Yes |

|    |                       |  |                                 |  |   |     |
|----|-----------------------|--|---------------------------------|--|---|-----|
| 17 | DAM                   | Services\Windows Services\ALTextCataloger\ALTextCataloger\TextCataloger.cs(496)                                  | DeleteFileFromTemporaryLocation | C:\SASMOM60_R14\MountPoint\OB\86581864861375\ACTUALFILE\SPA_Setup.txt.Original.20514857708265.TEXT | This is to cleanup temporary Folder while populating searchable alerts                  | Yes |
| 18 | DAM                   | Services\Windows Services\Cataloger\CatalogJobProcessor\JobProcessor.cs(736):                                    | NTFSJobCleanup                  | Catalog Jobpath  | To cleanup the temporary location after successful creation of ZIP file                 | Yes |
| 19 | DAM                   | Services\Windows Services\DAMJobCopy\DAMJobCopy\FWProcessor.cs(622):   | NTFSJobCleanup                  | Job path added in external cataloger mount point   | To cleanup the job folder after successful transfer of job to cataloger machine         | Yes |
| 20 | ProcessManager        | <Installation Dir>\Infrastructure\ProcessManager\ProcessManager\ProcessManagerClass_Ext2.cs(267)                 | ConsolidateDocument             | <Installation Dir>\MediaProcessor\<TenantName>\<UniqueID>  | Use to delete Files Created for Single file Consolidation                               | Yes |
| 21 | ProcessManager        | <Installation Dir>\Infrastructure\ProcessManager\ProcessManager\ProcessManagerClass_Ext6.cs(574)                 | ConsolidateDocument_MultiFile   | <Installation Dir>\MediaProcessor\<TenantName>\<UniqueID>  | Use to delete Files Created for Multi file Consolidation                                | Yes |
| 22 | AL_WorkbenchComponent | <Installation Dir>\Services\QueueComponents\AL_WorkbenchComponent\AL_WorkbenchComponent\AL_WorkbenchComponent.cs | ClearWebDAVLocation             | <WebDADVFolderPath>\<TenantName>\<ProcessID>   | Use to Delete Folder Containing PDF file for WebDav Sharing (Commenting)                | Yes |
| 23 | AL_WorkbenchComponent | <Installation Dir>\Services\QueueComponents\AL_WorkbenchComponent\AL_WorkbenchComponent\AL_WorkbenchComponent.cs | ClearWebDAVLocation             | <WebDADVFolderPath>\<UniqueID>   | Use to Delete hash Folder create by WebDav Service after Comments added by the reviewer | Yes |

|    |                    |  |                             |   |  |     |
|----|--------------------|--|-----------------------------|---|--|-----|
| 24 | Replication Agent  | Services\Queue Components\Repl icationAgent\Repl icationAgent\Repl icationAgent.cs(17 4)           | updateNTFS                  |   | We call this API to delete temporary backed up folder. | Yes |
| 25 | Replication Agent  | Services\Queue Components\Repl icationAgent\Repl icationAgent\Repl icationAgent.cs(74)             | updateNTFS                  |   |  | Yes |
| 26 | FileSystem Manager | Infrastructure\File Access Layer\FileSystem Manager\FileSyste mManager\FileSys temManager.cs(5 49) | RollBackOperat ions         | [Tenant MountPoint]\SHADOW_ FOLDER\Transaction Id | Clean up of shadow folder                              | Yes |
| 27 | FileSystem Manager | Infrastructure\File Access Layer\FileSystem Manager\FileSyste mManager\FileSys temManager.cs(5 56) | RollBackOperat ions         | [Tenant MountPoint]\SHADOW_ FOLDER\Transaction Id | Clean up of shadow folder                              | Yes |
| 28 | FileSystem Manager | Infrastructure\File Access Layer\FileSystem Manager\FileSyste mManager\FileSys temManager.cs(7 55) | CommitOperati ons           | [Tenant MountPoint]\SHADOW_ FOLDER\Transaction Id | Clean up of shadow folder                              | Yes |
| 29 | FileSystem Manager | Infrastructure\File Access Layer\FileSystem Manager\FileSyste mManager\FileSys temManager.cs(7 62) | CommitOperati ons           | [Tenant MountPoint]\SHADOW_ FOLDER\Transaction Id | Clean up of shadow folder                              | Yes |
| 30 | FileSystem Manager | Infrastructure\File Access Layer\FileSystem Manager\FileSyste mManager\FileSys temManager.cs(9 91) | ExecuteInDoub tTransactions | [Tenant MountPoint]\SHADOW_ FOLDER\Transaction Id | Clean up of shadow folder                              | No  |

|    |                           |  |  |  |                              |     |
|----|---------------------------|--|--|--|------------------------------|-----|
| 31 | FileSystem Manager        | Infrastructure\File Access Layer\FileSystem Manager\FileSystemManager\FileSystemManager.cs(998)                      | ExecuteInDoubleTransactions                        | [Tenant MountPoint]\SHADOW_FOLDER\Transaction Id | Clean up of shadow folder    | No  |
| 32 | WindowsAssetManager       | Infrastructure\File Access Layer\WindowsAssetManager\WindowsAssetManager\WindowsFileSystem.cs(316)                   | DeleteFolder                                       | [Tenant MountPoint]\FOLDER ID                    | Deleting empty directory     | Yes |
| 33 | WindowsAssetManager       | Infrastructure\File Access Layer\WindowsAssetManager\WindowsAssetManager\WindowsFileSystem.cs(346)                   | DeleteFolder(String FolderPath, Boolean Recursive) | [Tenant MountPoint]\FOLDER ID                    |                              | Yes |
| 34 | WindowsRemoteAssetManager | Infrastructure\File Access Layer\WindowsRemoteAssetManager\WindowsRemoteAssetManager\WindowsRemoteFileSystem.cs(345) | DeleteFolder(String FolderPath)                    | [Tenant MountPoint]\FOLDER ID                    |                              | Yes |
| 35 | WindowsRemoteAssetManager | Infrastructure\File Access Layer\WindowsRemoteAssetManager\WindowsRemoteAssetManager\WindowsRemoteFileSystem.cs(380) | DeleteFolder(String FolderPath, Boolean Recursive) | [Tenant MountPoint]\FOLDER ID                    |                              | Yes |
| 36 | WindowsRemoteAssetManager | Infrastructure\File Access Layer\WindowsRemoteAssetManager\WindowsRemoteAssetManager\WindowsRemoteFileSystem.cs(382) | DeleteFolder(String FolderPath, Boolean Recursive) | [Tenant MountPoint]\FOLDER ID                    |                              | Yes |
| 37 | XMLManager                | Infrastructure\XMLManager\XMLManager\XMLManager.cs(553)  | DeleteXML(String TypeName, ulong ObjectID)         | [Tenant XML Mount Point]\Object type\Object Id   | Delete XML files for objects | Yes |

|    |                     |  |  |   |   |     |
|----|---------------------|--|--|---|---|-----|
| 38 | XMLManager          | Infrastructure\XMLManager\XMLManager.cs(651)                       | DeleteXML(String TypeName)                                       | [Tenant XML Mount Ponint]\Object type\Object Id                         | Delete XML files for objects  | Yes |
| 39 | XMLManager          | Infrastructure\XMLManager\XMLManager.cs(1353)                      | CommitOperations(XmlNodeList LogNodes)                           | [Tenant XML Mount Ponint]\SHADOW_FOLDER\Transaction Id                  | Clean up of shadow folder   | Yes |
| 40 | XMLManager          | Infrastructure\XMLManager\XMLManager.cs(1469)                      | RollBackOperations(XmlNodeList LogNodes)                         | [Tenant XML Mount Ponint]\SHADOW_FOLDER\Transaction Id                  | Clean up of shadow folder   | Yes |
| 41 | XMLManager          | Infrastructure\XMLManager\XMLManager.cs(1583)                      | ExecuteInDoubtTransactions(XmlNodeList LogNodes, String strInfo) | [Tenant XML Mount Ponint]\SHADOW_FOLDER\Transaction Id                  | Clean up of shadow folder   | No  |
| 42 | MissingFile Utility | InstallationFolder\MFRU  | Shadow Folder Cleanup  | <Installation Directory>\<Mountpoint>\SHADOW_FOLDER\<FolderID>\<FileID> | To clean up shadow folder files and folders that are older than the buffer days | Yes |
| 43 | SessionManager      | Infrastructure\SessionManager\SessionManager\SessionManagerClasses | SignOut  | Temp\<SessionID>  | To clean up the temp folder   | Yes |

### 3.3. Installation Test Description:

No Installation is required as it is a manual file replacement on top of 6.0R14HF3. Six - machine setup is used for this testing.

## 4. Recommendation:

The SAS MOM 6.0 R14 HotFix04 can be released for general availability.

## 5. References & Credits:

### 5.1. References:

1. 60R14\_Hotfix04.xlsx
2. MOM 6.0 R 14 Hotfix 4.docx

### 5.2. Credits (MOM Testing Team):

1. Khalid Karim – Senior Manager – Testing, STG
2. Deepak Chourasia - Software Specialist – Testing, STG
3. Abhishek Kamble – Software Engineer – Testing, STG
4. Pankaj Jadhav – Senior Software Engineer – Testing, STG
5. Ganpati Revankar – Senior Software Engineer – Testing, STG
6. Ramnik Gahrotra – Software Specialist – Testing, STG
7. Surekha Devi – Software Engineer – Testing, STG
8. Harshada Phirke – Senior Software Engineer – Testing, STG
9. Pragati Doshi – Senior Software Engineer – Testing, STG
10. Talib Mukadam – Associate Testing – Testing, STG
11. Debashish Nath – Software Specialist – Testing, STG