

EMC[®] Documentum[®] Composer

Version 6.6

Release Notes

EMC Corporation
Corporate Headquarters:
Hopkinton, MA 01748-9103
1-508-435-1000
www.EMC.com

Copyright©2008 - 2010 EMC Corporation. All rights reserved.

Published May 2010

EMC believes the information in this publication is accurate as of its publication date. The information is subject to change without notice.

THE INFORMATION IN THIS PUBLICATION IS PROVIDED AS IS. EMC CORPORATION MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WITH RESPECT TO THE INFORMATION IN THIS PUBLICATION, AND SPECIFICALLY DISCLAIMS IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Use, copying, and distribution of any EMC software described in this publication requires an applicable software license.

For the most up-to-date listing of EMC product names, see EMC Corporation Trademarks on EMC.com.

All other trademarks used herein are the property of their respective owners.

Revision History:

May 2010: Release 6.6

Table of Contents

Chapter 1	Product Description	7
Chapter 2	New Features and Changes	9
	Composer Project and DAR file versioning.....	9
	XML application editor	9
	Performance improvements.....	9
	Logging improvements	9
	Overwrite on import	9
	Improved error handling	10
	Composer packages TCMReferenceProject	10
	BAM Reports are imported when importing a BAM Dashboard	10
	Change in behavior when installing Workflow Templates	10
Chapter 3	Fixed Problems	13
	Fixed problems.....	13
	Aspect artifacts require at least one attribute	13
	Installation options for non-root SysObjects.....	14
	Placeholder in smart container artifact must have at least one link	14
	Object name attribute values cannot contain any quotes	14
	Installing a DAR into a 5.3 repository requires a pre-install script	14
	JarDef artifacts and ModuleDef downloadable attachments might not get downloaded to the user's system	14
	After updating a JarDef that has already been installed in the repository, the updated JarDef is not downloaded to the user's system.....	14
	Process Templates and private Form Adapters are sometimes installed to the wrong locations	15
	TBO module can be installed without a corresponding object type	15
	When using Composer to import a TaskSpace application, Docview Forms is not implicitly imported.....	15
	Importing TaskSpace applications with the emc.importArtifacts Ant task is not supported	15
	Composer prompts you to save a type that has not been edited	16
	Members of existing groups or roles are inadvertently deleted	16
	Existing Process Templates in a repository are paused when a versioned copy of the same Process Template is installed to the repository with Composer.....	16
	Installing a project or DAR that contains a replica lifecycle fails	16
	Custom types that have a default value set for an attribute cannot have attached lifecycles	17
Chapter 4	Environment and System Requirements	19
	Software requirements	19
	Composer environment	19
	Supported product	20

Embedded products	20
Chapter 5 Known Problems and Limitations	21
Known problems	21
Renaming Composer projects causes problems in rare cases	21
Copying Composer messages to Eclipse ILog can result in poor Composer performance.....	21
Module editor allows attaching SysObject without content.....	22
If a content file contains “[i]” at the end of its filename, it cannot be removed in the method editor	22
The Scope drop down menu does not contain default scope values for aspect types	22
Type names and attribute data types cannot be changed after it has been installed	22
Modified artifacts do not get updated in the repository if their identities are changed	22
Copying and pasting a previously installed artifact is not supported	23
The version build information is difficult to find in headless Composer	23
Removing and adding a workflow to a Composer project fails.....	23
Importing structured data types from a DocApp results in an error	23
Dynamic workqueues are not imported when importing a TaskSpace application.....	24
Old JAR Definition content files are sometimes not deleted	24
Composer does not warn users that type attribute names that are too long	24
Dynamic work queues that are used in Process templates are not archived in TaskSpace DAR files.....	24
Composer allows duplicate artifact names.....	24
Lifecycles cannot be saved with the Save As command	24
DocApps improperly import types with names that begin with ‘dmc’	25
When using Composer 6.6 to import a BAM report from a D6.5 SP2 repository, run the following DQL update statements and apply a BAM D6.5 SP2 cumulative hotfix	25
There is confusion when users attempt to use old workspaces with newer versions of Composer (CMPSR-1779)	26
Hotfixes not included in this release.....	26
Limitations.....	26
Composer does not support replica artifacts	26
Migrating DocApps that contain BPM artifacts requires a 6.5 or later migration repository	27
Saving changes in permission set editor fields requires users to tab out	27
Composer supports only one content page for archived processes	27
Ant tasks cannot be run when Headless Composer is installed into a path that has spaces	27
A Composer project cannot be installed with headless Composer unless all referenced projects are in the installation workspace	27
Copying an artifact from one Composer project to another might not copy related artifacts.....	27
Reimporting a BPM artifact into a Composer project is not supported	28
Composer does not support multiple Process templates in a folder when importing from Process Builder.....	28
Some Composer 6.5 SP1 or earlier projects must be rebuilt to be installed.....	28
A Composer project cannot have duplicate type names.....	28
Custom types that have a default value set for an attribute cannot have attached lifecycles.....	28
Attaching a default aspect to a dm_folder type is not supported	29

Chapter 6	Technical Notes	31
	Composer project requirements.....	31
	Creating lightweight or shareable object types requires a license.....	31
	Running Ant tasks with headless Composer.....	31
	TaskSpace deployment	32
Chapter 7	Documentation	33
	Documentum Composer documentation set.....	33
	Obtaining the correct documentation	33
	Documentation corrections and clarifications	33
	Minor differences between Composer online help and user guide.....	34
Chapter 8	Software Media, Organization, and Files	35
	Software Media	35
	Organization	35
	Files.....	35
Chapter 9	Installation	37
	Installing Composer	37
	Installing the lightweight SysObject plugin	38
	Installing headless Composer	38
Chapter 10	Troubleshooting and Getting Help	41

List of Tables

Table 1.	Composer environment	19
----------	----------------------------	----

Product Description

EMC Documentum Composer provides a unified development platform to assemble, configure, and deploy EMC Documentum applications. Based on the Eclipse Integrated Development Environment, Composer provides a unified environment to create and maintain EMC Documentum application elements such as workflow templates, lifecycle definitions, aspects, object types, and DFS services.

The Eclipse plug-in model supports the broader ecosystem of customers, partners, and systems integrators. Developers assemble applications from components they develop or reuse from others, which reduces the time and cost to build vertical applications. This also enables organizations to configure rather than code, helping them to become more agile and responsive to market changes.

New Features and Changes

Composer Project and DAR file versioning

Composer Projects and Composer DAR files now have a version stamp that contains the version of Composer that generated the Composer Project or DAR file. For example, in a Composer project, the `version.properties` file contains the version string (such as 6.6.0.108).

Note: Newer versions of Composer cannot use workspaces created by an older version of Composer. Create a workspace first and then import projects from the old workspace into the new workspace.

XML application editor

Previously, you could only edit an existing XML application after importing the XML application into Composer. Composer 6.6 has support for creating XML application artifacts, as well as editing metadata for existing XML application artifacts.

Performance improvements

The performance of importing artifacts into Composer and installing artifacts to a repository has been improved.

Logging improvements

Composer has improved logging capabilities. For example, when installing a Composer project, the log file will include the types of artifacts being installed.

Overwrite on import

Composer can import artifacts from a repository and overwrite any existing artifacts. In previous versions of Composer, the import process would result in an additional copy of the artifact.

Note: Composer does not perform garbage collection on child artifacts. Composer imports dependant artifacts, but does not delete them from your project if they are no longer needed. For example, if you imported a workflow template that depended on e-mail templates, Composer imports the e-mail templates automatically. If some e-mail templates are later removed from the workflow template

and you do a re-import of the workflow template to the same Composer project, Composer does not automatically remove the e-mail templates from the project. Because of the lack of garbage collection, the recommended best practice for re-importing a TaskSpace Application into a Composer Project is to create a new Composer Project, rather than re-import updated artifacts in to an existing Composer Project.

Improved error handling

Composer has enhanced error checking capabilities that notify you of errors earlier and with more descriptive messages. For example, when installing a Composer Project that contains a Workflow template, Composer checks that Process Engine is installed to the repository. Another example of improved error handling is checking that a Composer Project or DAR file is valid.

Composer packages TCMReferenceProject

You no longer have to download, import, and reference the TCMReferenceProject manually. TCMReferenceProject is packaged with Composer, much like DocumentumCoreProject. You can safely use or extend any artifacts that are contained in TCMReferenceProject without downloading the project.

BAM Reports are imported when importing a BAM Dashboard

Before Composer 6.6, when you imported a BAM Dashboard, you had to also manually import related BAM reports. In Composer 6.6, when importing a BAM Dashboard, the related BAM reports are imported automatically.

Change in behavior when installing Workflow Templates

Before Composer 6.6, when installing a Workflow Template, Composer always installed it to the location specified by the installation parameter. For example, if you specified the installation parameter location to be /System/Applications/A during one installation, then changed it to /System/Applications/B during a subsequent installation, the Workflow Template is installed to two different locations.

In Composer 6.6, Composer looks for previous installations of a Workflow Template. If the Workflow Template is installed, Composer overwrites the existing instance even if you manually changed the installation parameter to a different location.

Fixed Problems

Within two weeks of the release, EMC will post a list of the fixed problems in the release on the Powerlink website (<http://powerlink.EMC.com>).

To view the list of fixed problems:

1. Log into the Powerlink website.
You must have a software support agreement to log in and access the list of fixed bugs.
2. From the Powerlink menu bar, navigate to **Support > Knowledgebase Search > Documentation and White Papers Search**.
The Knowledgebase Search page appears.
3. In the **Search For** text box, type **Fixed**.
4. In the **Refine by Product** text box, begin typing in the product name. Select your product from the list that appears.
5. Select the product **Version** for which you would like to view fixed problems.
6. From the **Select Document Type(s)** list box, select **Fixed Bugs List**.
7. Click **Search**.
8. Scroll to the bottom of the page and select the fixed problems document from the list of search results.

Fixed problems

This section identifies problems that have been resolved and were previously documented in the release notes as Known Problems or Limitations. A full list of the fixed problems for this release can be found in Powerlink.

Aspect artifacts require at least one attribute

Composer allows creating an aspect artifact without configuring any attributes for the aspect. However, if the project contains an aspect artifact without attributes, the installation of a project into a repository fails.

Installation options for non-root SysObjects

The installation options for owner and ACL artifacts should be configured in a post-install script.

Placeholder in smart container artifact must have at least one link

When adding a placeholder to a smart container, it must be linked at least one other smart container element. A placeholder cannot be an orphan element.

Object name attribute values cannot contain any quotes

If the project contains object name attribute values with quotes, the installation of a Composer project fails.

Installing a DAR into a 5.3 repository requires a pre-install script

When installing a DAR into a 5.3 repository, the DAR must have a pre-install script, even if it is empty. If the DAR does not contain a pre-install script, Composer throws a `NullPointerException` error.

JarDef artifacts and ModuleDef downloadable attachments might not get downloaded to the user's system

It is recommended that you install JarDef artifacts and ModuleDef downloadable attachments with the *version* option. If you install the JarDef with the *overwrite* option, your updated JAR file might not get downloaded to the user's machine. See [After updating a JarDef that has already been installed in the repository, the updated JarDef is not downloaded to the user's system](#), page 14 for more information.

After updating a JarDef that has already been installed in the repository, the updated JarDef is not downloaded to the user's system

If you update a JarDef, such as rewriting a method in a JAR file, and reinstall the JarDef, the updated JAR file sometimes is not downloaded to the user's machine (if you install the JarDef

with the “overwrite” option). If this happens, you need to clear the BOF cache for the JAR file to be downloaded. See [JarDef artifacts and ModuleDef downloadable attachments might not get downloaded to the user’s system, page 14](#) for more information.

Process Templates and private Form Adapters are sometimes installed to the wrong locations

If you import a Process template from the repository and change the installation location of the Process template, the child activities are not automatically updated with the new location. Set the locations of the child activities as well. For example, the correct installation location should be in System/Forms, but Composer might install them to System/Applications/*MyApp*.

Private Form Adapter files are also installed to the wrong location. Manually set the installation location to be in the Adapters subdirectory of the default installation location. You can set the location in the Composer project. For example, if the default location is System/forms/*MyForm*, change the location to System/forms/*MyForm*/Adapters.

TBO module can be installed without a corresponding object type

Composer currently does not prevent users from installing a TBO module without a corresponding object type in a repository. However, a TBO module cannot exist without a corresponding object type.

When using Composer to import a TaskSpace application, Docview Forms is not implicitly imported

This problem occurs when you import a TaskSpace application that contains a Task Form with an associated Docview Form. If only the Task Form is added to the TaskSpace application, the Docview Form is not implicitly imported. To also import the Docview Form, you must explicitly add it to the TaskSpace application.

Importing TaskSpace applications with the emc.importArtifacts Ant task is not supported

TaskSpace applications are not imported correctly when using headless Composer and Ant tasks. To import TaskSpace applications, use the Composer UI.

Composer prompts you to save a type that has not been edited

This problem occurs when you switch tabs (such as from the Display to the General tab) in the Type editor. If you save the type, change tabs, but do not make any changes, Composer detects the type as being modified and prompts you to save changes when closing the editor. If you are certain that you have not made any changes, you can cancel the message.

Members of existing groups or roles are inadvertently deleted

Creating and installing a new group or role that has the same name as an existing group or role in the repository causes the members of the existing group or role to be deleted. This happens when the installation upgrade option is set to **Overwrite Matching Objects**, which is set by default. To prevent this problem from happening, set the installation upgrade option for the group or role artifact to **Ignore Matching Objects**.

To set the installation upgrade option to Ignore Matching Objects:

1. Right-click the group or role artifact in the Documentum Navigator pane of Composer and select **Properties**.
2. Click **Document Artifact** on the left side of the **Properties** window.
3. For the **Upgrade** option drop down menu, select **Ignore Matching Objects**.

Existing Process Templates in a repository are paused when a versioned copy of the same Process Template is installed to the repository with Composer.

If you are installing a versioned Process Template to a repository that already contains a version of that Process Template, the existing instances of the original Process Template are paused and the Process template is put into the Draft state.

Installing a project or DAR that contains a replica lifecycle fails

The following error message is returned when you install a project or DAR that contains a replica lifecycle:

```
[DM_FOREIGN_E_REPLICA_GLOBAL]error: "Attempt to locally modify a global attribute of a foreign object. Attribute dm_policy.r_definition_state cannot be updated on replica object 468ab9b38000710b."; ERRORCODE: 100; NEXT: null
```


Custom types that have a default value set for an attribute cannot have attached lifecycles

If you created a custom type and assign a default value set for an attribute for that type, you cannot attach a lifecycle to that type. When installing the project, the installation fails.

Environment and System Requirements

Documentum Composer runs inside the Eclipse Integrated Development Environment (IDE). It is a Java application, and therefore requires a Java Runtime Engine (JRE). The Composer package includes a full copy of Eclipse and all required plug-ins. We do not explicitly support use of Composer plug-ins in an arbitrary Eclipse environment.

Software requirements

This section provides information on supported software environments.

The tables in this section reflect the latest versions of third-party products, upon which the EMC Documentum product depends, that are supported at the time of this release. For information on currently supported environments and future updates, refer to Product Information System on the Powerlink website: (<http://powerlink.EMC.com>). For detailed information about the policies governing supported products, refer to the *Customer Guide to EMC Software Support Services* available on the Powerlink website: (<http://powerlink.EMC.com>).

Composer environment

The Composer environment consists of a specific combination of operating system and processor. The following table lists the supported platforms for Composer.

Table 1. Composer environment

Operating system	Processor
Windows XP SP3 (32-bit version)	IA-32, x64
Windows Vista SP2(32-bit version)	IA-32, x64
Windows Server 2003 R2 with SP2 (32-bit version)	IA-32, x64
Windows Server 2003 SP2 (32-bit version)	IA-32, x64

Operating system	Processor
Windows Server 2008 R2 (32-bit version)	IA-32, x64
Windows 7 (32-bit version)	IA-32, x64

Supported product

Documentum Composer requires a Java Runtime Engine (JDK). You must install the Sun JDK before you install Composer. Composer supports the following JDK:

- Sun JDK 5.0 Update 22

Embedded products

Installing Documentum Composer includes the following additional products.

- Eclipse 3.4
- DFC D6.6
- DFS D6.6

Known Problems and Limitations

This section identifies problems and limitations that may affect your use of the product.

Note: This section and the Technical Notes section may refer to platforms or features that are not supported for this release of your product. Check [Chapter 4, Environment and System Requirements](#) to verify requirements.

EMC Documentum makes the latest information about customer-reported issues and known problems are posted on the Powerlink website (<http://powerlink.EMC.com>). You must have a software support agreement to log in and access the list of issues.

Known problems

This section describes known defects in EMC Documentum software that may affect your use of the product.

Renaming Composer projects causes problems in rare cases

Avoid renaming Composer projects. Renaming a project can cause Eclipse to crash in certain circumstances.

Copying Composer messages to Eclipse ILog can result in poor Composer performance

Turning on the **Copy Messages to Eclipse ILog** option causes Composer to respond slowly.

Module editor allows attaching SysObject without content

There is currently no validation check in the module editor to prevent users from attaching a SysObject without content when creating or modifying a module.

If a content file contains “[i]” at the end of its filename, it cannot be removed in the method editor

When using the method editor, an exception is thrown when you try to remove a content file with [i] at the end of the filename (for instance, Sample.java[1]). To prevent a file from being named this way, do not create a method, delete it, and create another method with the same content file and same name as the one you deleted.

The Scope drop down menu does not contain default scope values for aspect types

When you create an aspect type, the Scope drop down menu in the Display tab does not display default scope values (webtop and DCDesktopClient) for aspects. Specify your own values in the drop-down menu.

Type names and attribute data types cannot be changed after it has been installed

To rename a type after you have installed it, copy and paste the type, rename the newly pasted copy, and delete the old type from your project. You can also create a type from scratch with the new name and delete the old type. If you only change the name of the existing type without copying and pasting the type or creating a type, Composer tries to update the type in the repository by referencing it with its new name. This results in an error because the type still has its old name in the repository.

The same is true if you change an attribute’s data type. To change the attribute’s data type, create an attribute and delete the old attribute.

Modified artifacts do not get updated in the repository if their identities are changed

If you change an artifact’s identity, such as its installation location or name, and reinstall the project, the artifact with the new identity is never created or upgraded. If other changes were made, the artifact with the old identity is updated instead. If you want to update an artifact’s identity, create an artifact instead.

Copying and pasting a previously installed artifact is not supported

If you copy and paste an artifact that has been installed to a repository, modify the copied artifact, and re-install the project, Composer does not install two instances of the artifact. Composer installs the original artifact then overwrites that artifact with the copy and pasted version. This does not occur if you copy and paste an artifact that has not been installed to the repository.

The version build information is difficult to find in headless Composer

You can find version information for headless Composer in the `ComposerHeadless\plugins\com.emc.ide.external.dfc_1.0.0\documentum.config\version.properties` file.

Removing and adding a workflow to a Composer project fails

The following scenario results in an import failure:

1. Import a workflow that was built in Process Builder into a Composer project.
2. Make changes to the workflow in Process Builder.
3. Delete the workflow from the Composer project and re-import the workflow to pick up the new changes.
4. The import fails.

To avoid this problem, create a Composer project if you want to re-import a modified workflow.

Importing structured data types from a DocApp results in an error

If you are trying to import a DocApp from a repository or from a DocApp archive and the DocApp contains a structured data type, the Composer project will not compile. Composer incorrectly names the structured data types with a `dm_wfsd` prefix, which is not allowed.

To correct the error, delete the types that have the `dm_wfsd` prefix or import the types directly from the repository (not with the DocApp).

Dynamic workqueues are not imported when importing a TaskSpace application

When importing TaskSpace applications with process templates that contain dynamic workqueues, the workqueues are not imported. To have the workqueues imported into the Composer project, import them into the project manually.

Old JAR Definition content files are sometimes not deleted

When you associate a content file with a JAR Definition, and then specify a new content file, a copy of the old content file is retained along with the new content file.

Composer does not warn users that type attribute names that are too long

Type attribute names can be 27 characters long. If you specify a name that is longer, Composer does not notify you until you try to install the project or DAR file.

Dynamic work queues that are used in Process templates are not archived in TaskSpace DAR files

If you are importing a TaskSpace application that contains a Process template that uses dynamic workqueues, the dynamic workqueues do not get imported. Manually import the dynamic workqueues into the Composer project.

Composer allows duplicate artifact names

When you create an artifact that is the same type and name as another artifact, Composer appends a “[n]” to the end of the original artifact and gives the new artifact the given name. This convention can cause confusion as to which artifact is the correct one to use, so do not use duplicate names when possible.

Lifecycles cannot be saved with the Save As command

When modifying or creating lifecycles, do not use the **File > Save As** command when saving the lifecycle. The lifecycle does not save properly.

DocApps improperly import types with names that begin with 'dmc'

When importing a DocApp into a Composer project, you can encounter errors because types with names that begin with 'dmc' are imported. These special types cannot exist in user Composer projects. You can safely remove the 'dmc' artifacts from your project to eliminate the errors. The following is a list of 'dmc' artifacts that can get imported:

- dmc_wfsd_element_boolean.lwtype
- dmc_wfsd_element_date.lwtype
- dmc_wfsd_element_double.lwtype
- dmc_wfsd_element_integer.lwtype
- dmc_wfsd_element_string.lwtype
- dmc_wfsdrp_boolean.lwtype
- dmc_wfsdrp_date.lwtype
- dmc_wfsdrp_double.lwtype
- dmc_wfsdrp_integer.lwtype
- dmc_wfsdrp_string.lwtype

When using Composer 6.6 to import a BAM report from a D6.5 SP2 repository, run the following DQL update statements and apply a BAM D6.5 SP2 cumulative hotfix

In the Documentum 6.6 timeframe, the Business Activity Monitor (BAM) server changed the schema for one of the underlying objects. To successfully import a BAM Report from a pre-6.6 repository, run the following DQL update query:

```
alter type bami_rg_report_entity_field add fixed_order integer

go

update bami_rg_report_entity_field objects set fixed_order=column_order

go
```

After you run the DQL update query, also apply the latest BAM D6.5 SP2 cumulative hotfix.

To use Composer to import a BAM report from a pre-D6.5 SP2 repository, use Composer D6.5 SP2 plus the Composer D6.5 SP2 cumulative hotfix.

There is confusion when users attempt to use old workspaces with newer versions of Composer (CMPSR-1779)

Newer versions of Composer cannot use workspaces created by an older version of Composer. Create a workspace first and then import projects from the old workspace into the new workspace.

To create a workspace and import projects:

1. To create a workspace:
 - a. Navigate to **File > Switch Workspace > Other....**
 - b. Specify a location for the new workspace.
 - c. Click **OK**.
2. To import existing projects (or projects in the old workspace) to a new workspace:
 - a. Navigate to **File > Import....**
 - b. Select **Documentum / Existing Projects Into Workspace**.
 - c. Select the directory of the existing project (or directory of the old workspace).
 - d. (Optional) Select **Copy projects into workspace** to copy projects to the workspace directory.
 - e. Click **Finish**.

Ensure that all files in the existing projects are writable. Composer can upgrade any project file. The *Documentum Composer User Guide* provides information on importing projects.

Hotfixes not included in this release

This release does not include hotfixes created after March 31, 2010 for previous releases of the product. Please contact EMC Customer Support for more information.

Limitations

This section describes limits on the usability of current functionality. The limitations may be part of the product design or may result from issues with associated third-party products.

Composer does not support replica artifacts

Composer cannot differentiate between regular objects and replicated objects. In a case where both, the regular and the replicated object point to the same object, such as an alias set, Composer tries to update both the regular and the replicated object. Since replicated objects are read-only objects, Composer cannot modify the replica and the DAR installation fails.

Migrating DocApps that contain BPM artifacts requires a 6.5 or later migration repository

Migrating a DocApp that contains BPM artifacts into a Composer project fails with migration repositories before version 6.5. DocApps with BPM artifacts require a version 6.5 or later migration repository.

Saving changes in permission set editor fields requires users to tab out

If a user modifies a field in the permission set editor, the user must use the Tab key and tab out of the field or the changes are not saved.

Composer supports only one content page for archived processes

When a process with more than one page is archived using Composer and the archived process is then installed into a repository, only the first content page of the process gets installed.

Ant tasks cannot be run when Headless Composer is installed into a path that has spaces

Install headless Composer into a path with no spaces. Ant version 1.7, which is packaged with Composer, does not behave correctly when installed in a path that contains spaces.

A Composer project cannot be installed with headless Composer unless all referenced projects are in the installation workspace

If you are installing a project with headless Composer, all reference projects must be in the installation workspace.

Copying an artifact from one Composer project to another might not copy related artifacts

Copying artifacts from one project to another might not copy all related artifacts or resources.

Reimporting a BPM artifact into a Composer project is not supported

You cannot import a BPM artifact into a Composer project and then try to re-import the same artifact into the same project.

Composer does not support multiple Process templates in a folder when importing from Process Builder

If you are using Process Builder to create Process templates and their corresponding activities, do not create multiple Process templates in the same folder. Composer cannot import the process templates if there are multiple ones in the same folder. Use a different folder for each Process template.

Some Composer 6.5 SP1 or earlier projects must be rebuilt to be installed

If you have a Composer project or DAR file that was built with Composer 6.5 SP1 or earlier and are trying to install it with Composer or the DAR Installer 6.5 SP2 or later, the installation fails in the following circumstances:

- The Composer project contains one or more Folder artifacts that contain either a subfolder or artifacts.
- The Composer project contains a Folder artifact that has its Owner, Security, or Upgrade option (unless it is set to Overwrite) specified in the properties page for the artifact.

To remedy the situation, import the Composer project into Composer 6.5 SP2 or later and rebuild it. You can now install the project from Composer or install the generated DAR file with the DAR Installer.

A Composer project cannot have duplicate type names

Composer does not allow duplicate type names in a single Composer project. If you try to install a project that has duplicate type names, the following error occurs:

```
precondition dfTypeObject must be null before creating a new type object
```

Custom types that have a default value set for an attribute cannot have attached lifecycles

If you created a custom type and assign a default value set for a custom attribute for that type, you cannot attach a lifecycle to that type. When installing the project, the installation fails.

Attaching a default aspect to a dm_folder type is not supported

Do not attach a default aspect to any dm_folder types in the repository. If you do, additional aspects for the dm_folder type cannot be installed with Composer.

Technical Notes

This section provides configuration and usability notes for current product features.

Composer project requirements

Every Composer project must have a **src** folder, even if the **src** folder is empty. If the **src** folder is deleted from a project, it is not recreated automatically when the project is build and Composer throws a validation error. If you get a validation error, add a **src** folder to your project and rebuild the project.

Creating lightweight or shareable object types requires a license

A Documentum High-Volume Server license is required to create a lightweight or shareable object type. The license must be specified on the machine that hosts Content Server.

Running Ant tasks with headless Composer

When building a project with Headless Composer using the **emc.importProject**, **emc.build**, and **emc.dar** Ant tasks, it is recommended to always start with importing the original project distribution into a newly created workspace to avoid corruption of the workspace.

TaskSpace deployment

When preparing a TaskSpace DAR for redeployment, for example to a production environment, you must manually import any custom folders that were created in the application.

Documentation

This section describes the documentation related to this product.

Documentum Composer documentation set

These documents comprise the Composer documentation set for this version of the product:

- *Documentum Composer User Guide*, version 6.6
- *Documentum Composer Release Notes*, version 6.6 (this document)
- *Documentum Composer Quick Start Guide*, version 6.5 SP3
- *BOF Development Tutorial*, version 6.5 SP3
- *Building a Documentum Application Tutorial*, version 6.5 SP2 (P/N 300-009-464)
- *Introduction to Composer for DAB users*, version 6.5 SP2 (P/N 300-009-174)

Obtaining the correct documentation

The Powerlink website (<http://powerlink.EMC.com>) contains the downloadable packages for specific product versions along with the release notes, installation guide, and other relevant documentation associated with the product version. To locate product documentation, navigate to Support > Technical Documentation and Advisories > Software ~ D ~ Documentation, then select the product name and version number.

Note: Most of the Content Management products are listed under Software D > Documentum ?, where ? = a letter, or letters, in the alphabet.

Product documentation that is available online from the application (such as online help) does not appear as a separate item. It is automatically downloaded and installed with the software.

Documentation corrections and clarifications

This section lists corrections to the product documentation.

Minor differences between Composer online help and user guide

There are minor differences between the Composer online help and the *Composer User Guide*. The *Composer User Guide* has the most up-to-date information.

Software Media, Organization, and Files

This section describes the media in which the software is available, the organization of the product components in the available media, and the file names for all available product components which can be downloaded.

Software Media

This product is available as an FTP download from the Powerlink website (<http://powerlink.EMC.com>). You should have received instructions through E-mail regarding how to download products.

Organization

The Powerlink website (<http://powerlink.EMC.com>) provides access to <https://EMC.subscribenet.com/control/dctm/index>, which provides a complete listing of the products available for download.

Files

The following modules/files comprise the contents of this release:

- *Documentum Composer Version 6.6 User Guide* (pdf)
- *Documentum Composer Version 6.6 Release Notes* (this document)
- *Documentum Composer Quick Start Guide Version 6.5 SP3* (pdf)
- *BOF Development Tutorial, version 6.5 SP3*
- *Building a Documentum Application Tutorial Version 6.5 SP2* (pdf)
- *Introduction to Composer for DAB Users Version 6.5 SP2* (pdf)

- **DCTM_Composer_R.6.6** (zip)
- **DCTM_Headless_Composer_win_R.6.6** (zip)

Installation

Composer is shipped in two versions, a UI-based version with wizards, dialogs, and editors, and a non-UI command line version that includes a set of Ant tasks for each of the common Composer functions, such as import, build, and install. The non-UI version of Composer is also referred to as headless Composer.

EMC Documentum also offers a lightweight SysObject plug-in for Composer that must be installed separately.

Note: Currently, only applications designed for Documentum High-Volume Server can make proper use of lightweight objects. Documentum High-Volume Server is an extension of Documentum Content Server that supports features implemented to solve common problems with large content stores, such as E-mail archiving. It requires an additional license key specified when Content Server is installed. For more information about lightweight object types and Documentum High-Volume Server, see *EMC Documentum High-Volume Server Developer Guide*.

This section describes how to install Documentum Composer, headless Composer, and the lightweight SysObject plug-in. Ensure that your system meets the requirements listed in [Chapter 4, Environment and System Requirements](#) before installing the software.

Installing Composer

This section describes how to install UI-based Composer.

To install Composer:

1. Download the **DCTM_Composer_<version>.zip** file from the download site: (<http://powerlink.EMC.com>).
2. Extract the .zip file to a directory on your local drive.
3. Configure the connection broker, as follows:
 - a. Change to the **..\Composer\plugins** directory.
 - b. Double-click the **com.emc.ide.external.dfc_1.00** folder.
 - c. Double-click the **documentum.config** folder.
 - d. Open the **dfc.properties** file with a text editor, such as Notepad. Add the DFC and connection broker information, similar to the following:

```
dfc.docbroker.host[0]=[DocBroker IP address or host name]
```

- e. Save your changes.
4. Navigate to the ...**Composer** directory and click the **eclipse** icon to start Composer.

Installing the lightweight SysObject plugin

The lightweight SysObject plugin is not part of the main Composer distribution and must be installed in the <Composer_root>/plugins directory after you have installed Composer.

To install the lightweight SysObject plugin:

1. Download the **LightweightObject_<version>.zip** file from the download site (<http://powerlink.EMC.com>).
2. Extract the plugin to the same directory as Composer. For example, if you extracted Composer to the C:\ directory, you also must extract the **LightweightObject_<version>.zip** file to the C:\ directory.
3. Change to the <Composer_root>/plugins directory and verify that the com.emc.ide.artifact.lwdclass_1.0.0.jar and com.emc.ide.artifact.lwdclass_ui_1.0.0.jar files are in the directory.

Installing headless Composer

This section describes how to install headless Composer.

To install headless Composer:

1. Download the **DCTM_Headless_Composer_<platform>_<version>.zip** file from the download site: (<http://powerlink.EMC.com>).
2. Unzip the .zip file to a directory on your local drive. The path name of that directory must not contain any spaces. If the installation directory for headless Composer contains spaces, the Ant tasks may not run correctly.
3. Configure the connection broker, as follows:
 - a. Change to the ..**ComposerHeadless\plugins** directory.
 - b. Double-click the **com.emc.ide.external.dfc_1.00** folder.
 - c. Double-click the **documentum.config** folder.
 - d. Open the **dfc.properties** file with a text editor, such as Notepad. Add the DFC and connection broker information, similar to the following:

```
dfc.docbroker.host[0]=[DocBroker IP address or host name]
```

- e. Save your changes.

Troubleshooting and Getting Help

Documentum's technical support services are designed to make your deployment and management of Documentum products as effective as possible. The *EMC Customer Support Guide* provides a thorough explanation of Documentum's support services and policies. You can download this document from the Powerlink website (<http://powerlink.EMC.com>) by navigating to: **Support > Request Support > Customer Support Guide**.