Model Manager PLUS Installation and Administration for 15.50

Contents

Legal notices

Restricted rights legend

CoCreate Software, Inc
3801 Automation Way #110
Fort Collins, CO 80525 U.S.A.

Notice

1. Introduction

This document shows you how to install and configure Model Manager PLUS.

Model Manager PLUS is a product from CoCreate's Consulting organization that boosts your Model Manager product installation with powerful add-on functionality and tools.

CoCreate

2. Installation of Model Manager PLUS

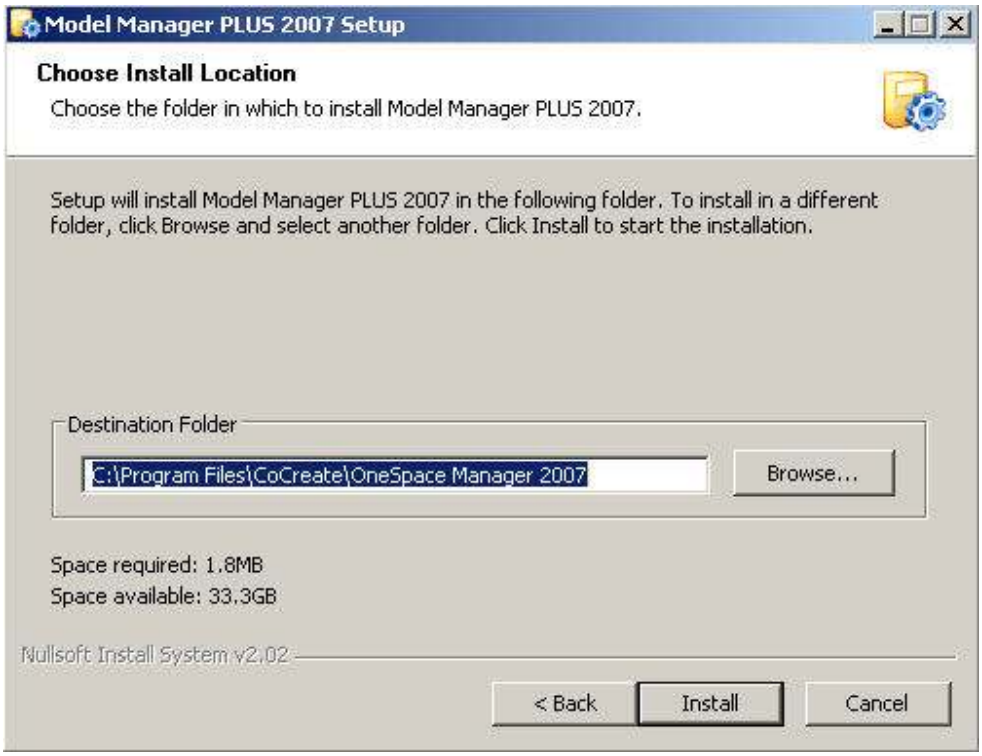
Prerequisite: You must have a running installation of OneSpace Manager 2007 **15.50** to install Model Manager PLUS. This module is not compatible with older versions of OneSpace Manager or Work Manager!

2.1 Installation

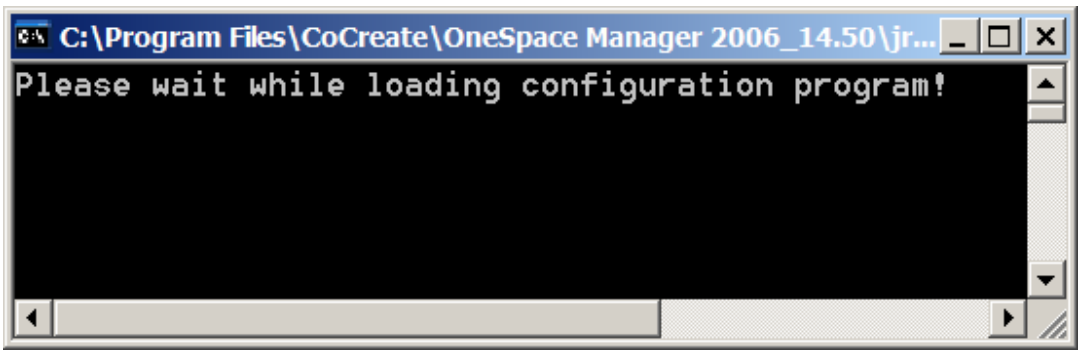
1. To start the installation, run **`ModelManagerPlusSetup.exe`** on the server with the OneSpace Manager 2007 15.50 installation**.**

CoCreate

Choose the install location of OneSpace Manager 2007



The Model Manager PLUS Setup dialog will appear:



CoCreate

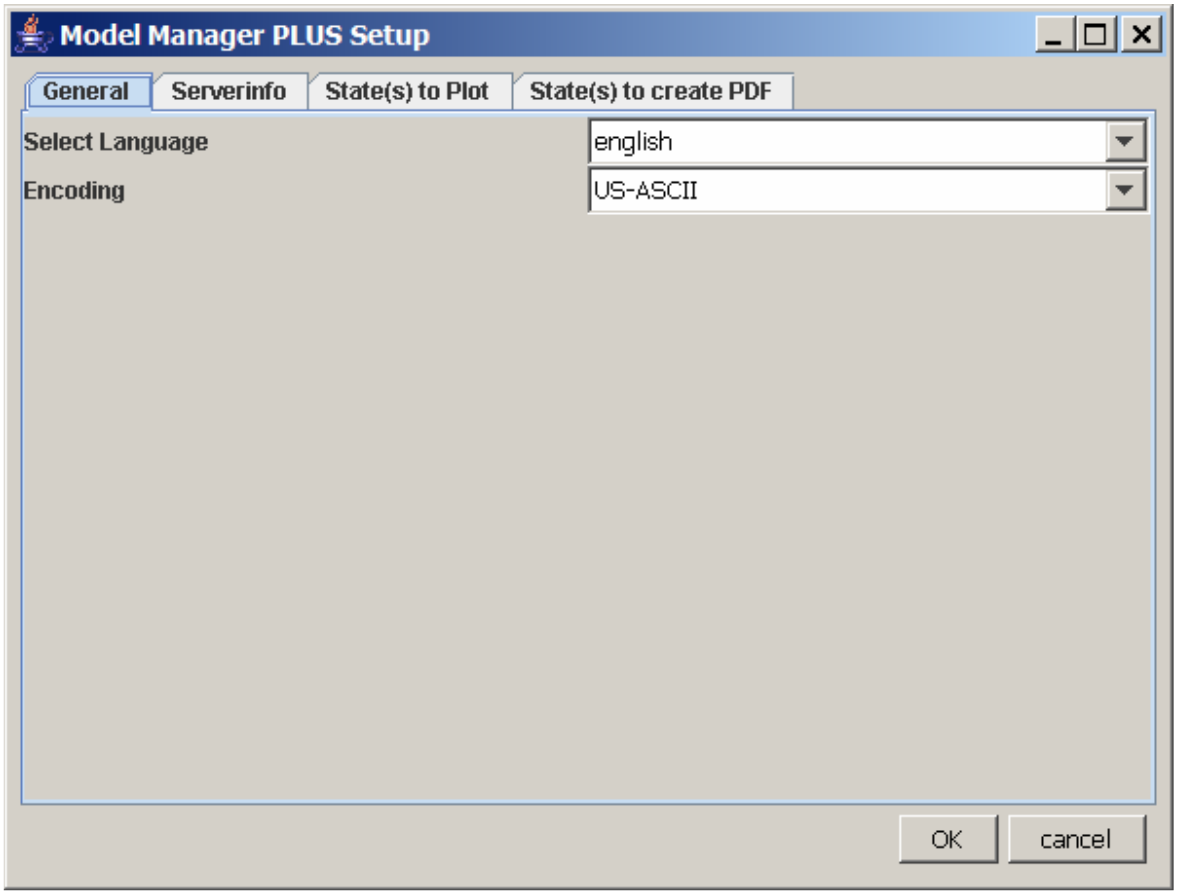
Model Manager PLUS requires configuration information about your OneSpace Manager installation. If a **`custom.xml`** file exists, most of this information will be read automatically and the message below will be displayed:

If you do not have a **`custom.xml`** file, you'll need the following information to proceed:

- Installation path and port of your database server.
- Installation path and port of your fileserver.
- Installation path of your license server.
- Installation path Software Distribution server.
- Installation path Notification Mail Server.
- Installation path Notification Mail Server Email.

This information will be necessary to create the startup files for **Model Manager PLUS**, **Print Server,** and **Output Server**.

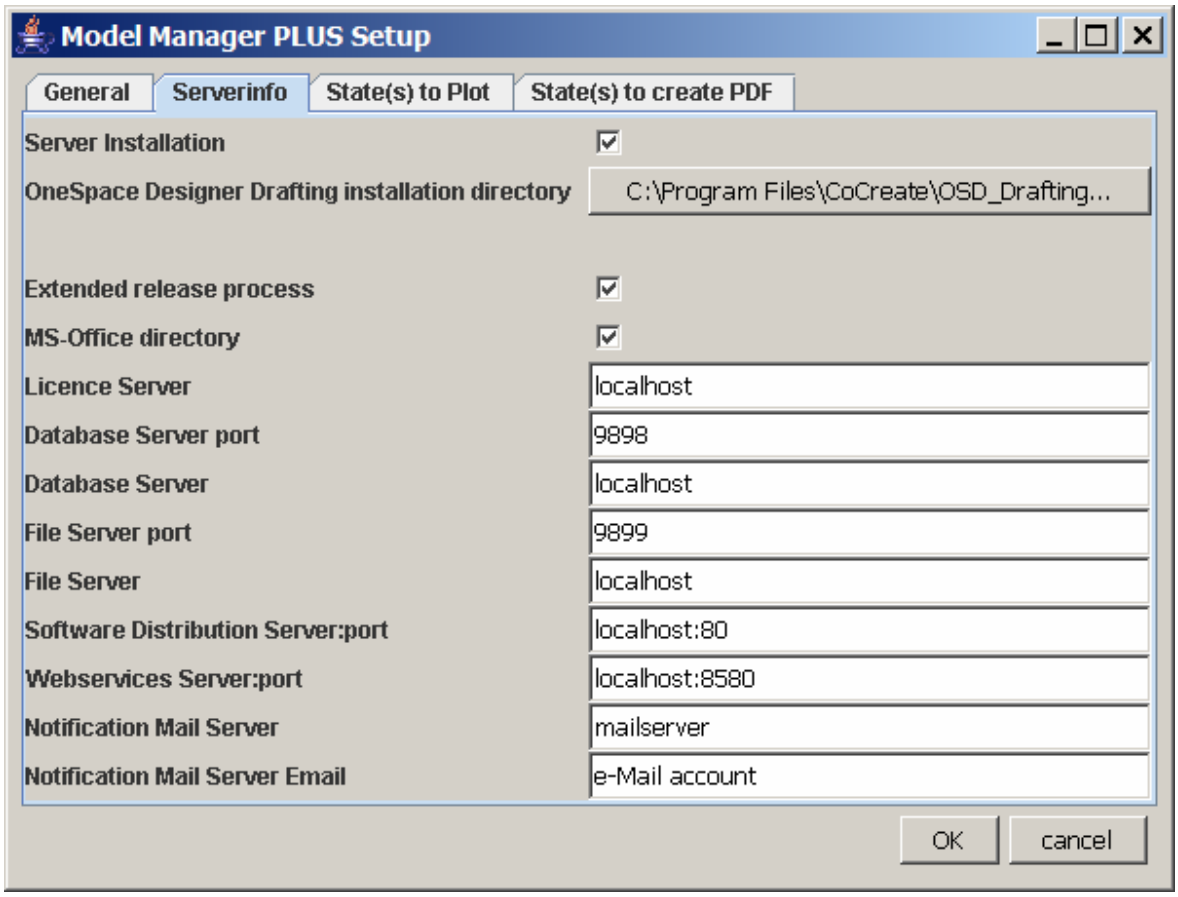
2. The Model Manager PLUS Setup dialog will appear:



Select the language and encoding

During the installation process of Model Manager Plus the files **`mmplus.xml`** and **`ops.xml`** are generated. You can choose the encoding of the generated XML files from a ComboBox during the installation process.

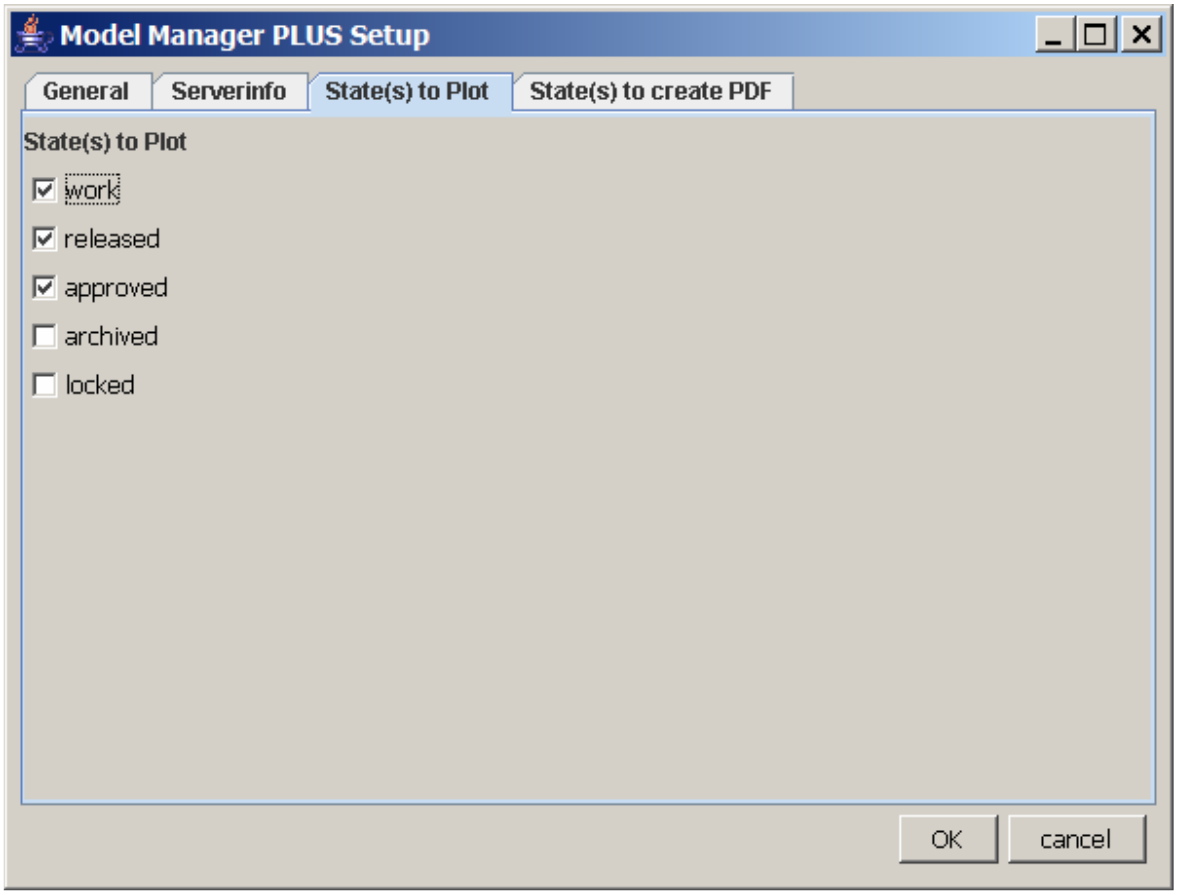
3. Enter server information



- Check the **Server Installation** checkbox.
- Select the directory of OSD Drafting
- Check the **Extended release process** checkbox
- Check the **MS- Office Integration** checkbox

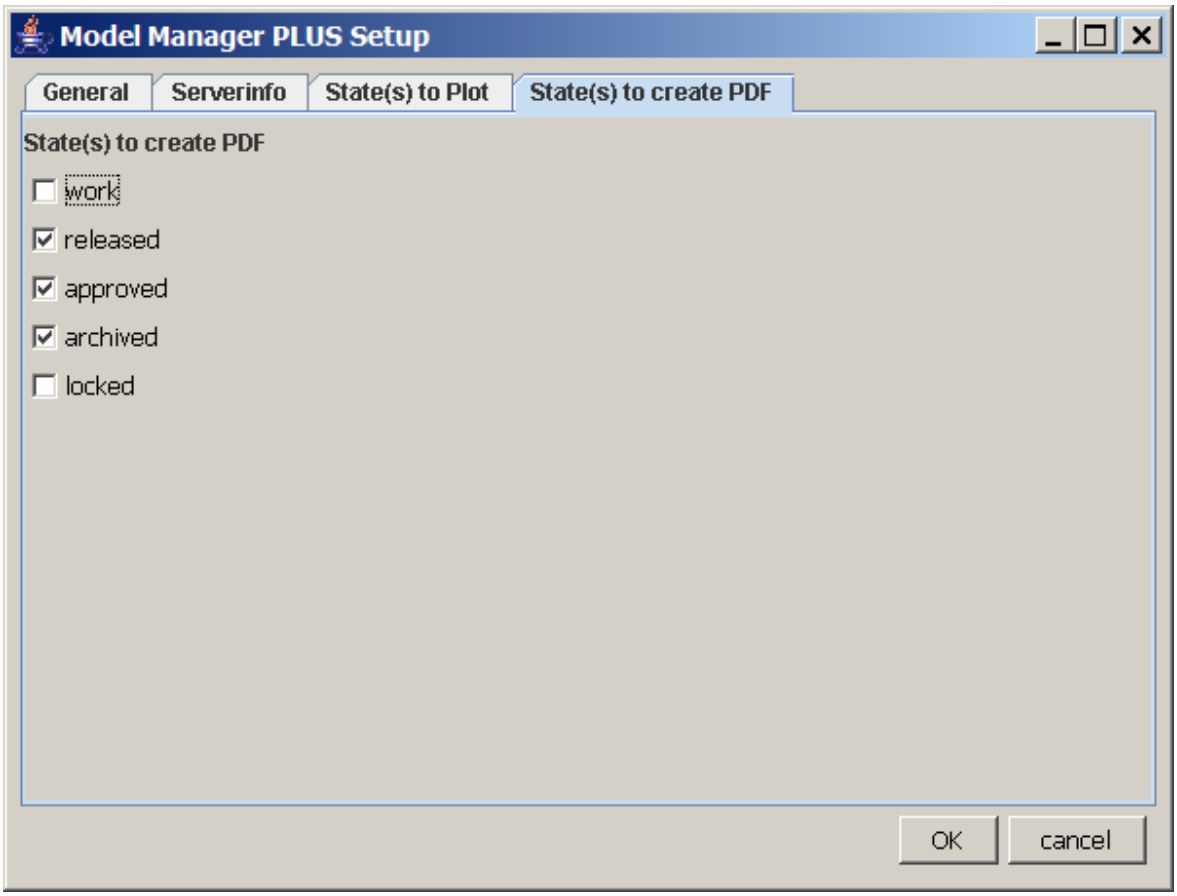
CoCreate

4. State(s) to Plot



Check the **State(s) to Plot** checkboxes

5. State(s) to create PDF



Check the **State(s) to create PDF** checkboxes

Check all entries and click **OK** if correct.

6. Edit
`C:\Program Files\CoCreate\OneSpace Manager2007\config\plot\plot.ini`
to define the printers used by the Plot Manager. See Section 5.2, ***Printer configuration for Plot Manager*** for additional information.

7. Create a new Model Manager installer (**ModelManagerSetup.exe**) to integrate Model Manager PLUS by running

```
C:\Program Files\CoCreate\OneSpace Manager 2007\clntwin\mmbuild.bat
```

in an MSDOS window. This will take several minutes. Once it finishes, the new **ModelManagerSetup.exe** will be available on the Software Distribution Server.

8. Run **ModelManagerSetup.exe** by accessing the Software Distribution Server web page and selecting the **Install Model Manager** link.

9. Create the classes necessary for Office Integration, the Output Server, and the Description Catalog by logging in to Model Manager as a member of the group **sysadmin** as follows:

    1. Start Model Manager
    2. Enter the login and password of a member of the sysadmin group.
    3. Select **mmplus** from the **Select Configuration** pick list. Click **OK**. The classes will be created as part of startup. You may check for their successful creation by selecting **File > New > Document**, and expanding the Class list. If MS_WORD is in the list, class creation was successful.

3 Office Integration

With Model Manager PLUS, you can manage Microsoft Word and Microsoft Excel documents in Model Manager.

3.1 Microsoft Word

The following chapters describe how to configure Model Manager PLUS to manage Microsoft Word documents.

3.1.1 Classes

The documents are managed in a separate Model Manager class. By default the following class will be installed:

MS_WORD Microsoft Word documents

In addition, the MS templates have a separate class:

MS_TEMPLATES Microsoft Templates

Business Object Class for the Office Integration:

com.mmplus.office.MmOffice

Within this Object class, different store options are defined, like “Create New Document,” “Create New Version,” and “Overwrite.”

For load purposes, the following Action Menu Class is defined:

com.mmplus.office.actionmenu.OfficeActionMenu

Within this class, a specific load routine for Office documents is defined. The load routine examines all documents and part attributes and transfers them to the Office application. Later, it checks if a file exists for this document.

Model Manager acts as a DDE Server and listens to the topic MMDdeServer and ModelmanagerThe DDE Server is started in the class MMDDEStarter. The registration of the loader class is done through the tag <CommandLoaderClass>:

com.mmplus.dde.MMDDEStarter

3.1.2 Templates

To check in templates:

1. Create an empty document in the class **MS_TEMPLATES**.
2. When creating the document, the Application has to be specified:
   WinWord Microsoft Word application
3. Check in the template file to the newly created document.

3.1.3 Installation

The Model Manager classes for the Office Integration are created the first time you start Model Manager PLUS after installation. See Section 2 for further information.

The message file, **mmoffice.msg,** contains the display messages. These must reside in the **lib/nls/msg** directories.

3.1.4 Configuration

The configuration of Office Integration is done within the XML file. The tags are defined within the <OfficeIntegration> tag.

<OfficeIntegrationActive>

0 The Office integration is inactive.
1 The Office integration is active.

<WordClassName>

MS_WORD Model Manager class name for Word documents.

<TemplateClassName>

MS_TEMPLATES Model Manager class name for templates.

<WordExePath>

`winword.exe`

<SysIdWord>

Winword DDE system identifier for Word.

<TemplateWinSizeX>

400 Size *x* in pixels for the Template window.

<TemplateWinSizeY>

300 Size *y* in pixels for the Template window.

<OfficeCommandWaitTime>

500 Time in ms before the command will be sent to the Office application.

<OfficeStartWaitTime>

2000 Time in ms before starting Office application

3.1.5 Configuring macro security

Word must access Visual Basic for Applications code to interface with Model Manager PLUS. Therefore, macro security must be set to **medium** in Microsoft Word so the macros in the templates can run. Follow these steps to set macro security:

1. Open Microsoft Word.
2. Select **Tools > Options > Security tab > Macro Security menu**.
3. On the Security dialog, go to the **Security Level** tab and change the Security level from **High** to **Medium**. Now each time you open a document in Model Manager PLUS you get the message **Disable Macros** or **Enable Macros**. Choose **Enable Macros** to get the Microsoft Application to work with Model Manager PLUS.

3.2 Microsoft Excel

The following chapters describe how to configure Model Manager PLUS to manage Microsoft Excel documents.

3.2.1 Classes

The documents are managed in a separate Model Manager class. By default the following class will be installed:

MS_EXCEL Microsoft Excel documents

In addition, the MS templates have a separate class:

MS_TEMPLATES Microsoft Templates

Business Object Class for the Office Integration:
com.mmplus.office.MmOffice

Within this Object class, different store options are defined, like “Create New Document,” “Create New Version,” and “Overwrite.”

For load purposes, the following Action Menu Class is defined:
com.mmplus.office.actionmenu.OfficeActionMenu

Within this class, a specific load routine for Office documents is defined. The load routine examines all documents and part attributes and transfers them to the Office application. Later, it checks if a file exists for this document.

Model Manager acts as a DDE Server and listens to the topic MMDdeServer and ModelMan. The DDE Server is started in the class MMDDEStarter. The registration of the loader class is done through the tag <CommandLoaderClass>:

com.mmplus.dde.MMDDEStarter

3.2.2 Templates

To check in templates:

1. Create an empty document in the class **MS_TEMPLATES**.
2. When creating the document, the Application has to be specified:
   Excel Microsoft Excel application
3. Check in the template file to the newly created document.

3.2.3 Installation

The Model Manager classes for the Office Integration are created the first time you start Model Manager PLUS after installation. See Section 2 for further information.

The message file, **mmoffice.msg,** contains the display messages. These must reside in the **lib/nls/msg/french** directories.

3.2.4 Configuration

The configuration of Office Integration is done within the XML file. The tags are defined within the <OfficeIntegration> tag.

<OfficeIntegrationActive>
0 The Office integration is inactive.
1 The Office integration is active.

<ExcelClassName>
MS_EXCEL Model Manager class name for Excel documents.

<TemplateClassName>
MS_TEMPLATES Model Manager class name for templates.

<ExcelExePath>
`excel.exe`

<SysIdExcel>
Excel DDE system identifier for Excel.

<TemplateWinSizeX>
400 Size *x* in pixels for the Template window.

<TemplateWinSizeY>
300 Size *y* in pixels for the Template window.

<OfficeCommandWaitTime>

500 Time in ms before the command will be sent to the Office application.

<OfficeStartWaitTime>

2000 Time in ms before starting Office application

3.2.5 Configuring macro security

Excel must access Visual Basic for Applications code to interface with Model Manager PLUS. Therefore, macro security must be set to **medium** in Microsoft Excel so the macros in the templates can run. Follow these steps to set macro security:

1. Open Microsoft Excel.
2. Select **Tools > Options > Security tab > Macro Security menu**.
3. On the Security dialog, go to the **Security Level** tab and change the Security level from **High** to **Medium**. Now each time you open a document in Model Manager PLUS you get the message **Disable Macros** or **Enable Macros**. Choose **Enable Macros** to get the Microsoft Application to work with Model Manager PLUS.

CoCreate

4 Release process

With **Model Manager Plus** you can verify that all assemblies or parts have reached a required state before others are allowed to change. The release process is configurable in that a company can decide if they want to release assemblies part by part or level by level.

Part by part. When the system is set to release part by part (<ReleaseLevel> = 0), Model Manager PLUS will prevent users from raising the state of an assembly when any of its parts remain at a lower state. For example, you cannot promote an assembly to “released” when one of its models is in the state “work.” See the examples in user Help for sample scenarios.

Level by level. When the system is set to release level by level (<ReleaseLevel>=1), Model Manager PLUS will not allow you to raise the state of an assembly when any of its subassemblies remain in a lower state. However, if an assembly contains only models, or if its subassemblies have already reached the higher state, Model Manager PLUS will allow you to raise the state of the assembly. It will also raise the state of associated models that have not yet been promoted. See the examples in user help for sample scenarios.

Models and drawings. When releasing the model, the system can be configured so that related drawings can also be released automatically.

The release process relies on the STATE_ORDER in the class DT_STATEFLOW. Therefore it is important when adding new states that the STATE_ORDER attributes in the existing DT_STATEFLOW elements are not changed.

4.1 Configuration

The configuration for the Release process is done within the XML file. The tags are defined within the <ReleaseProcess> tag.

<ReleaseLevel>

Optional:	indicates the rules for children parts. Defaults to 0.
0 or 1	A 0 ReleaseLevel requires that children part models be in the same state or higher then the new state for the parent assembly. A 1 ReleaseLevel will move the children part models to the new parent assembly state if the current state of children is the same as the current state of the parent assembly.

<AutoReleaseDrawing>

Optional	
True or False	Defaults to false. If true, any drawings owned by the parent model will have their state's moved to the new state of the parent.

<CheckDrawing>

Optional	
True or False	Defaults to true. If true, makes sure that the status of any drawings who's state will be change is valid. If its status is not valid, the state change will fail.

<StateOrders>
Required

States	One or more tags or a comma separated list that define the logical ordering of states from lowest to highest.

<StateAfterRelease>
Optional

State	Defaults to none. The state to move previously released Models and Drawings when a new version of the Models or Drawings is moved to the release state. If none, the previously released Models and Drawings will remain in the release state.

CoCreate

5 Plot Management

Plot Manager enables you to send drawings directly from Model Manager to the configured printers.

Plot Manager is configured in the following Model Manager PLUS files:

- Plot Manager is defined in **`mmplus.xml`**.
- Printers are defined in **`plot.ini.`**
- Variables for the frames are defined for the right database schema language in **`globals.m`**

5.1 XML- Plot Management configuration

The configuration of the Plot Manager is done within the XML file (**`ops.xml`** in the directory …**`\PrintServer\config_ops\config`**).
The tags are defined within the < PlotManager> tag.

```
<PlotManager>
        <PlotManagerActive>true</PlotManagerActive>
        <ClassName>com.mmplus.spooler.PlotDialog</ClassName>
        <GroupName>PLOTGROUP</GroupName>
        <MonitorGroup>MONITORGROUP</MonitorGroup>
        <PrintServerDirectory>PATH</PrintServerDirectory>
        <DraftingDirectory>PATH</DraftingDirectory>
        <ViewOption>-view</ViewOption>
        <PlotIniFileUserLocation>PATH</PlotIniFileUserLocation>
        <StateToPlot>STATES</StateToPlot>
        <SearchLevel>LEVEL</SearchLevel>
</PlotManager>
```

Configuration is defined within the <PlotManager> tag as follows:

<PlotManagerActive>*VALUE*</PlotManagerActive>
Defines whether Plot Manager shall be used or not.
Optional:

True	Plot Manager is active.
False	Plot Manager is inactive.
Empty	Plot Manager is inactive.

<ClassName>*Java class*</ClassName>
Required: Defines the Java class used for the Plot Manager.

com.mmplus.spooler.PlotDialog	Fully qualified path to the Java class.

If the class does not exist, the menu item **Send to plotter** will not appear

<GroupName>*GROUPNAME*</GroupName>
Optional: Defines the Model Manager group that can use the Plot Manager

PLOTGROUP	Plot Manager is only available to defined group.
Empty	Plot Manager available to all users

In the database, any group may be defined to enable users to use the Plot Manager.
If this tag does not exist, Plot Manager will be available to all users.

<Monitor Group>*GROUPNAME*</Monitor Group>
Optional: Defines the Model Manager group that can use the Plot Manager Monitor

GROUPNAME	Plot Manager Monitor is only available to defined group.
Empty	Plot Manager Monitor available to all users

In the database, any group may be defined to enable users to use the Plot Manager Monitor.
If this tag does not exist, Plot Manager Monitor will be available to all users.

<PrintServerDirectory>*PrintServer*</PrintServerDirectory>
Required: Defines the relative path where PrintServer is installed.

PATH	Path to install directory of PrintServer

<DraftingDirectory>*PATH*</DraftingDirectory>
Required: Defines the path where **OSD- Drafting** is installed.

PATH	Path to install directory of **OSD- Drafting**.

<ViewOption>-view</ViewOption>
Defines the command-line parameter passed to Designer Drafting when passed.

-view	2D Access will run instead of Drafting.
Empty	Drafting will run. Note. This setting is required to output file types other than pdf.

See the section ***File Export Configuration*** for more information on generation of other file types.

<PlotIniFileUserLocation>*PATH*</PlotIniFileUserLocation>
Optional: Defines the location of the local plot.ini file.

Path to local plot.ini	Points Plot Manager to this local plot.ini file.
Empty	ModelManager client uses the plot.ini file in the **`/config`** directory.

<StateToPlot>*STATE*</StateToPlot>
Optional: Defines which drawings states are allowed to plot.
There may be multiple tags with this name.

Work, approved, released	List of states of which drawings can be plotted.
Empty	Drawings in any state can be plotted.

<SearchLevel>*VALUE*</SearchLevel>
Optional: Defines the search level for drawings in structures.

0	Whole assembly structures.
1	One level of assembly structures.
Value	Any integer to define the query level.

5.2 Printer configuration for Plot Manager

The name of the configuration file is **`plot.ini`** and localized in the following directory:

`C:\Program Files\CoCreate\Shared\Model Manager 2007\config\plot`

Printers and plotters for the Plot Manager are defined here. You do not have to configure all available formats.

PRINTER

Define the printer as follows.

[PRINTER]

Computer=*localhost*

The variable “Computer” names the computer on which the PrintServer is installed.

e.g,. **`Computer=PrintServer`**.

If the printer at the print server is installed as a local printer, this entry remains empty

e.g., **`Computer=`**

ShareName=*HPG85XX*

Name of the printer. It must available in the network.

e.g., **`lj4mv_a`**

Alias=*HP Office Jet G85 in design department*

Name displayed in the selection list in Model Manager / Drawing Manager.

e.g., **`LaserJet 4MV Accounting`**

Format=*A3, A2, A1*

List of the formats printed by the printer. This list can be different from the formats actually supported by the printer. The formats are comma-delimited. *e.g., A4, A3*

The format may be mapped within ME10/Drafting to the formats corresponding to the printer's paper selection. This mapping is done by the macro **Sco_set_print_paper_size**. To use the correct format for each printer, this macro has to be adjusted accordingly. To do so, it's important to correctly spell the sheet formats each printer uses. For a DesignJet, an entry **A4** for the format is not sufficient. The entire string in the macro which is also offered during manual paper selection needs to be checked.

e.g., **`ISO A4 - 210 x 297 mm.`**

(The trailing period has to be entered as well!)

FormatAlias=*A3-sheet,A2-sheet,A1-sheet*

The name displayed in the selection list in Model Manager / Drawing Manager

Output=*PCL, HPGL, TIFF, JPG, JPEG*

The listing of output formats supported by the printer

Note. The variable *Output* is currently not supported.

MaxOutputSize=*0 0 180 250*

Dimension for the output size. Like in the file **plotdefs**

PARAMETERS

In this section global parameters are defined:

[PARAMETERS]

ReloadIni=*1*

1 Reload of “plot.ini” at each call of the plot menu is forced.

0 File “plot.ini” not reloaded

FitToPaper=*1*
The variable “FitToPaper” is related to the section [SCALE]. Scales output to fit paper size
1 “FitToPaper” is active for the drawing and defining a scale is **not** possible
0 Defining a scale is now active

DialogReduction=*100*
To define how much smaller (in pixels) the dialog window is compared to the monitor width.

UserCanSeeAllJobs=*1*
0 User can only see their own jobs in the Plot Manager monitor
1 User can see their own and all pending plot jobs in the Model Manager PLUS Plot Manager Monitor.

SCALE
In this section all user-allowed scaling can be defined:

[SCALE]
ScaleInit=1*:1,1:2,1:5,1:10,1:2.5,1:100,10:1,2.5:1,5:1*
A choice is only possible if fit to paper is inactive.

COLOROPTION
In this section the macro names are described that can be used for the drawing output.

To use this option the macros must input in OneSpace Manager Drafting. These macros are defined to get the right drawing output. For example:

```
[COLOROPTION]
PLOTMACRO=True color,Sco_true_color
PLOTMACRO=Pen width,Sco_pen_width
PLOTMACRO=Black and white,Sco_black_and_white
```

Pen size Macro name, must before input in Drafting
Pen size defined by color Show text in Plot Manager

ROTATE
In the section the possible rotation angles for print output are defined.

[ROTATE]
ANGLES=*0,90,180,270*
The Variable “ANGLES” contains a comma separated list.

5.3 Frames configuration for Plot Manager and Output Server

The configuration file **`globals.m`**, resides in the following directory:

`C:\Program Files\CoCreate\Shared\Model Manager 2007\PrintServer\`

The following (global) variables must be defined in this file:

- Switch to reload TB (Title block)
  - DEFINE Awmc_val_tb_reload TRUE END_DEFINE

- TB (Title block) – class name e.g. “FRAME_2D”
  - LET Awmc_val_tb_class 'FRAME_2D'

- Format, offset and version attribute name
  - DEFINE Awm_db_format "FORMAT" END_DEFINE
  - DEFINE Awm_db_offset “OFFSET” END_DEFINE
  - DEFINE Awm_db_version “VERSION” END_DEFINE

- Count of change notes to be displayed on the drawing
  - DEFINE Awmc_val_max_change_notes 3 END_DEFINE

- TB (Title block) prefix name e.g. “.tb_frame_”
  - DEFINE Awmc_val_tbname_prefix ".tb_frame_" END_DEFINE

6 Number Generator

The Number Generator provides generation of continuous part and model identification numbers, and automates generation of item designations using standard rules.

6.1 How to configure a rolling number generator

The administrator can configure a rolling number generator for attributes that can be edited, including model names, Masterdata names, and pseudo attributes.

The first step is to configure the attribute in the XML configuration file. You will need to use the following XML tags:

- <AddPreviewButton> - If true, the preview next number button is displayed, otherwise it is not displayed.
- <AllowUserOverwrite> - True if the user is allowed to change the value generated by the rolling number generator. False if the user is not allowed to change the value. The default is true. This tag is specific to the rolling number generator.
- <AttributeEditorClass> - Editor and renderer that Model Manager should use in the Save dialog for the attribute. Set to com.osm.datamgmt.editor.NumberGenerator.Editor
- <BusinessObjectClass> - Java object Model Manager should use when a new instance of the attribute is created or when an existing instance is opened. Set to com.osm.datamgmt.biz.RollingNumberGenerator.
- <EnabledDuringSave> - This tag is used when there is a rolling number generator on both the model name and Masterdata name. If true, the Masterdata name will be generated by its rolling number generator on a save instead of using the model name. If false, the Masterdata name generator will only be used when you create Masterdata with File > New > Masterdata. Put this tag in the Masterdata name attribute.
- <Format> - Defines formatting symbols to specify how the number should be displayed to the user (see Java Tutorial on Decimal Formats)
- <RollingNumberGeneratorName> - Name of the rolling number generator, which you will create in Model Manager.

Example

```
<Class extends="DMModel">
<Name>MODEL_3D</Name>
<NameFormat>%NAME%: [%Version%] %State%</NameFormat>
<Attribute>NAME
<BusinessObjectClass>com.osm.datamgmt.biz.RollingNumberGenerator</BusinessObjectClass>
<AttributeEditorClass>com.osm.datamgmt.editor.NumberGeneratorEditor</AttributeEditorClass>
<RollingNumberGeneratorName>modelnameRNG</RollingNumberGeneratorName>
<Format>00000</Format>
<AddPreviewButton>true</AddPreviewButton>
<AllowUserOverwrite>true</AllowUserOverwrite>
</Attribute>
</Class>
```

Once the rolling number is configured in the XML file, the administrator can create the rolling number generator in Model Manager.

1. Click **File > New > Document**.
2. Select **RN_Generator** in the **Select class** dialog.
3. Click **OK**.
4. Fill in the values in the **New Document** dialog

- **Current Value** – the value that the Rolling Number Generator will start with. This value must be an integer.
- **Increment** – amount that should be added for each sequential number. This value must be an integer.
- **Name** – name of the rolling number generator. This must match the name configured with the <RollingNumberGeneratorName> tag in the XML configuration file.
- **Postfix** – any characters that should follow the generated number
- **Prefix** – any characters that should go before the generated number

2. Click **OK**.

6.2 How to use the rolling number generator

See user help for details.

6.3 How to modify the rolling number generator

See user help for details.

6.4 A more advanced use of the rolling number generator

You can use any combination of characters and attribute values as the name of the rolling number generator in the XML configuration file. For example, if you set the name to %DB_STATE%, the state of the model will be substituted for the name of the rolling number generator.

When you create the rolling number generator in Model Manager you need to create a rolling number generator named for each state. For example, create a rolling number generator named “WORK” and another named “APPROVED.”

In addition, you can use any combination of characters and attribute values to define the Prefix to Postfix. For example, you can set the Prefix to be XYZ[%DB_STATE] and the value of the state attribute for the element will be substituted for %DB_STATE.

Note
The configured postfix and prefix strings should not contain a hyphen (‘-‘) if the prefix or postfix contains attributes. The hyphen character is used by the parser and may cause incorrect values.

---

7 Description Catalog

With the Model Manager PLUS Description Catalog or Thesaurus you can assign naming inside the OSDM and pass the values to Model Manager during the save process.

The input dialog for the Description Catalog inside OneSpace Designer Modeling can offer multi-language fields. It is possible to import these values from an external table.

7.1 XML- configuration for the Description Catalog

The Description Catalog is defined within the tag <DescCatalog>:

```
<DescCatalog>
        <Active>true</Active>
        <ClassName>DESCRIPTION</ClassName>
        <Attributes>DESCRIPTION,DESCRIPTION1,DESCRIPTION2,LANGUAGE</Attributes>
   <MapAttributes>DESCRIPTION,DESCRIPTION1,DESCRIPTION2,LANGUAGE</MapAttributes>
        <DbLength>255,255,255,255</DbLength>
        <DbDescription>Description,Description1,Description2,Language</DbDescription>
    </DescCatalog>
```

Tags work as follows:

<Active>VALUE</Active>
Optional: Controls whether the catalog is used or not.

True	Description catalog is active.
False	Description catalog is inactive.
Empty	Description Catalog is inactive.

<ClassName>DESCRIPTION</ClassName>
Required: Defines the name of the database class that contains the naming.

DESCRIPTION	Database- class for description catalog.

<Attributes>
Required: Defines the attributes of the description class.

DESCRIPTION,DESCRIPTION1,DESCRIPTION2,LANGUAGE	Comma-separated list of the attributes.

<MapAttributes>
Optional: Defines the attribute of the *target class 3d model*. They are mapped 1:1 from the description class.

DESCRIPTION,DESCRIPTION,DESCRIPTION,LANGUAGE	Comma-separated list of the attributes.

<DbLength>
Required: Length of attributes (used when class is created).

255,255,255,255	Length of the attributes in a comma-separated list.

<DbDescription>
Required: Describes the labeling of the attributes in the database.

Description,Description2,Description3,Language Attribute labels in a comma-separated list

You can name the description catalog class free. The attribute names can also be chosen class free.

Note
It is important always to use correct mapping: **<Attributes>** and **<MapAttributes>** .

If an existing database class is used for the Description Catalog, you have to configure it in the XML file.

```
<Class >
            <Name>DESCRIPTION</Name>
            <BusinessObjectClass>com.osm.biz.WMDoc</BusinessObjectClass>
            <CanCreateInUI>true</CanCreateInUI>
            <AuditTrail>OFF</AuditTrail>
            <EditorClass>com.osm.datamgmt.editor.BaseEditor</EditorClass>
            <NameFormat>%DESCRIPTION%:[%LANGUAGE%]</NameFormat>

            <Attribute>DESCRIPTION</Attribute>
            <Attribute>DESCRIPTION1</Attribute>
            <Attribute>DESCRIPTION2</Attribute>
            <Attribute>LANGUAGE</Attribute>
</Class>
```

Note
The variable "*language*" is currently not supported.

7.2 Configuration of the Description Catalog in OneSpace Designer Modeling

Besides the XML configuration, additional configuration needs to be done inside OneSpace Designer Modeling to activate this feature.

Basically you need to ensure that the files located in the directory
…**/CoCreate/Shared/Model Manager 2007/OSDM/modelmanager**
are loaded.

The file **sco_mm_load.lsp** in this directory contains the load statements for all other files that need to be loaded.

There are two ways to configure this:

1. If you have defined a SDCORPCUSTOMIZEDIR environment variable, you need to
   - Copy the whole modelmanager subdirectory from
     …**/CoCreate/Shared/Model Manager 2007/OSDM/modelmanager**
     into your corpcustomize directory

   - Add a load statement for the **sco_mm_load.lsp** file in, e.g., your **mm_customize** file
     e.g,

```
 (load (format nil "~a/modelmanager/sco_mm_load" (oli::sd-sys-getenv
"SDCORPCUSTOMIZEDIR")))
```

2. If you don't have the SDCORPCUSTOMIZE variable defined, you need to

- Add a load statement for the **sco_mm_load.lsp** file in, e.g., your **mm_customize file** (see the example below)
- Change the load statements in the **sco_mm_load.lsp** file.

Here's an example:

- Add the following load statement into the **mm_customize** file

```
(setf *mm-custom-path* "D:/Program Files/CoCreate/Shared/Model Manager
2007/OSDM/modelmanager")

(load (format nil "~a/sco_mm_load_new" *mm-custom-path*))
```

- Replace the existing load statements in the **sco_mm_load.lsp** file using the following code.

```
(sd-with-current-working-directory *mm-custom-path*
    (load "sco_mm_variables")
    (load "sco_mm_customize")
    (load "sco_mm_desc_lexikon")
    (load "sco_mm_dialog")
    (load "sco_mm_browser")
      )
```

8 Output Server

The Output Server (OPS) lets you generate several file types when triggered by state changes in Model Manager. It runs as a separate Model Manager client.

8.1 Starting the Output Server

There are two modes to start the OPS.

Loop mode: This mode is started by running **`runop.bat`**. In this mode the server queries the Output Server table and automatically processes jobs found in the table. The Model Manager workspace is not available, only a Model Manager-like command window will appear. The OPS will also stop automatically.

Debug mode: This mode is started by running **`runop_debug.bat`**. In this mode the server displays a Model Manager workspace and an OPS dialog to manage jobs manually.

When this class is created, the user **opsmgr** is created and added to the group **sysadmin**.

8.2 XML- Configuration for the Output Server

XML configuration for the OPS is performed in the file
…**`\CoCreate\Shared\Model Manager 2007\PrintServer\config_ops\config\ops.xml`**
on the machine where the OPS is running.

The tags that affect the Model Manager Plus client like <StateToWork> and <OutPutServerTableName> also need to be configured on the clients in the file **`mmplus.xml`**.

The Output Server is defined by the tag <OutPutServer>.

Possible tags inside the section <OutPutServer> follow:

```
<OutPutServer>
        <Active>true</Active>
        <OutPutServerTableName>OPS_TABLE1</OutPutServerTableName>
        <StateToWork>released</StateToWork>
        <OutPutServerShutDownAt>time<OutPutServerShutDownAt>
        <OutPutServerShutDownAllowed>true</OutPutServerShutDownAllowed>
        <OutPutServerWaitTime>10</OutPutServerWaitTime>
        <OutPutServerLog>false</OutPutServerLog>
        <OPSDisplayFormat>dd.MM.yyyy</OPSDisplayFormat>
        <OPSKillDraftingTime>300</OPSKillDraftingTime>
        <OPSKillDraftingMacro>Macro name</OPSKillDraftingMacro>
        <OPSStoreFileTypes>file types</OPSStoreFileTypes>
        <OPSStoreDwgVersion>ACAD Version</OPSStoreDwgVersion>
        <OPSStoreDirectory>PATH</OPSStoreDirectory>
        <CreateMultiDXF>true</CreateMultiDXF>
        <CreateMultiDWG>true</CreateMultiDWG>
        <CreateMultiDWF>true</CreateMultiDWF>
        <CreateMultiPDF>true</CreateMultiPDF>
        <CreateMultiIGES>true</CreateMultiIGES>
        <CreateMultiSVG>true</CreateMultiSVG>
</OutPutServer>
```

Tags work as follows:

<Active>*VALUE*</Active>

Required: Defines whether the OPS is used or not.

True	OPS is used.
False	OPS is not used.
Empty	OPS is not used.

<OutPutServerTableName>*TABLENAME*</OutPutServerTableName>

Required: Defines the OSM class for the OPS. OPS jobs are written to this table.

OPS_TABLE1	DB-Class name for the Output server.

<StateToWork>*released*</StateToWork>

Required: Defines the state that, when transitioned to, will cause the OPS to generate a PDF. If the state subsequently changes from this state, the generated PDF document will automatically be deleted.

There may be multiple tags with this name.

<OutPutServerShutDownAt>*time string*<OutPutServerShutDownAt>

Time string	Time string to shut down OPS, e.g., “22:30”

<OutPutServerShutDownAllowed>*true*</OutPutServerShutDownAllowed>

True	OPS will shutdown if shutdown time has arrived.

<OutPutServerWaitTime>*10*</OutPutServerWaitTime>

Integer	Seconds to wait before executing next job query.

<OPSDisplayFormat>*dd.MM.yyyy*</OPSDisplayFormat>

Time format	Time format to be used for drafting (TB references) time strings, e.g., CREATED_AT

<OPSKillDraftingTime>*300*</OPSKillDraftingTime>

Integer	Seconds to wait before Drafting will be killed.

<OPSKillDraftingMacro>*Macro* name</OPSKillDraftingMacro>

Macro name	Macro that should be used to kill Drafting.

<OPSStoreFileTypes>*file types*</OPSStoreFileTypes>

CSL	File types to be stored during release process.

<OPSStoreDwgVersion>*ACAD Version*</OPSStoreDwgVersion>

Macro option	Macro option to store DWG files.

<OPSStoreDirectory>*PATH*</OPSStoreDirectory>

Path	Path to export files to the file system

<CreateMulti???>*true*</CreateMulti???>

True	Every sheet will be stored in a separate file.

Replace the ??? with one of the following file types:

- DXF
- DWG
- DWF
- VRML
- SVG
- IGES

8.3 File export configuration

The Output Server can create and store different file formats and check the files into the database.

Note
To enable generation of the following file types, the <ViewOption> tag in the <PlotManager> section of the **`ops.xml`** file must be set to empty. See section ***5.1 XML-Plot Management configuration*** for more.

Do the following to configure this option:

- Modify the tag **<OPSStoreFileTypes>*file types*</OPSStoreFileTypes>** Enter any of the following file types (comma-separated) Drafting is able to create.
  - DXF
  - DWG
  - DWF
  - VRML
  - SVG
  - IGES

  If you want to create DWG files (AutoCAD file type) you also have to define the desired AutoCAD version. Enter the store version in the tag **<OPSStoreDwgVersion>*ACAD Version*</OPSStoreDwgVersion>**. For more information about the possible AutoCAD versions see the OneSpace Designer Drafting documentation.

For each file type you can define a different store location. The store root directory is defined in the tag **<OPSStoreDirectory>*PATH*</OPSStoreDirectory>**

- Directory names for export additional formats (DXF, DWG…)
  - DEFINE Sco_ops_dxf_extern_dir "dxf_export" END_DEFINE
  - DEFINE Sco_ops_pdf_extern_dir "pdf_export" END_DEFINE
  - DEFINE Sco_ops_vrml_extern_dir "vrml_export" END_DEFINE
  - DEFINE Sco_ops_dwf_extern_dir "dwf_export" END_DEFINE
  - DEFINE Sco_ops_svg_extern_dir "svg_export" END_DEFINE
  - DEFINE Sco_ops_iges_extern_dir "iges_export" END_DEFINE
  - DEFINE Sco_ops_dwg_extern_dir "dwg_export" END_DEFINE

The created files can be imported into the database.

- Switch to check in the created formats
  - DEFINE Sco_ops_checkin_neutral_formats TRUE END_DEFINE