

Automatic Registration of Frames, BOM Table Templates, and Position Number Templates in CoCreate Model Manager

Contents

Legal notices	2
Restricted rights legend	2
Notice	2
Introduction	3
Audience	3
Typographical conventions.....	3
Registration Processes	4
CoCreate Annotation	4
CoCreate Drafting.....	5
CoCreate Drafting Frame Registration commands in <i>ui_toolbar.m</i>	6
CoCreate Annotation Frame Registration commands in <i>am_wm_customize</i>	6
Un-registering the default CoCreate Annotation frames.....	7
How to turn off the automatic registration processes.....	8
Customizing CoCreate Annotation BOM registration:	8
Locale Filters	9
Default Filter Definitions.....	9
How to override <i>wmconf.xml</i> default filters.....	11
Sample Name Restriction Filter Definition	11
Restricting frame and bom_flag element registration by controlling database elements	12
Writing Custom Java Filter Classes.....	12
Database Schema Changes for the new LOCALE code	12
How to find more information about locale filters	13
Java filter classes	13
XML Tags related to Registration Processes and Locale Filters	15
Step by Step Instructions	16
How to add custom frames to the database.....	16
How to add custom CoCreate Drafting BOM templates to the Database.....	17
Basic Steps for Controlling CoCreate Annotation Frame Registration.....	18
Basic Steps for Controlling CoCreate Drafting Frame Registration	20
Basic Steps for Controlling CoCreate Annotation BOM Template Registration.....	22
Basic Steps for Controlling CoCreate Drafting BOM Template Registration	24
How to control if frames are loaded or saved with drawings:	26

Legal notices

All rights reserved. Reproduction, adaptation, or translation without prior written permission is prohibited, except as allowed under copyright laws.

Restricted rights legend

Use, duplication, or disclosure by the U.S. Government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and Software clause in DFARS 252.227-7013. Rights for non-DOD U.S. Government Departments and Agencies are as set forth in FAR 52.227-19(c)(1,2).

Parametric Technology GmbH
Edisonstrasse 8
85716 Unterschleissheim
Germany

Parametric Technology Corporation
140 Kendrick Street
Needham, Massachusetts 02494
USA

Notice

The information contained in this document is subject to change without notice. CoCreate makes no warranty of any kind with regard to this manual, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. CoCreate shall not be liable for errors contained herein or for incidental or consequential damages in connection with the furnishing, performance, or use of this material.

UNIX® is a registered trademark of the Open Group. Microsoft® and MS-DOS® are U.S. registered trademarks of Microsoft Corporation. Netscape and Netscape Navigator are U.S. trademarks of Netscape Communications Corporation. Windows® and Windows NT® are trademarks of Microsoft Corporation.

Introduction

Beginning with CoCreate Model Manager 2006 and CoCreate Drawing Manager 2006, Frames, BOM Table Templates, and Position Number Templates are automatically registered in CoCreate Drafting and CoCreate Annotation. The registered elements will be filtered based on language locale filters.

Audience

Intended readers of this document should have a full understanding of system architecture, databases, PDM engineering tasks and issues on working in a distributed environment.

Typographical conventions

The following typographical conventions are used in this documentation:

Italic is used for

- Directory locations, filenames, document names, and program names.
- Internet domains, network addresses and URLs.
- New terminology, where it is defined.

Bold is used for

- Names of windows, buttons, menu items and other elements on the screen.

Fixed-width is used for

- Commands and options that should be typed exactly as they appear.
- Tags and parameters in markup languages, such as HTML or XML, and configuration files.

Note You will also find helpful notes, warnings and technical tips highlighted in special boxes like this one throughout this document.

Registration Processes

CoCreate Annotation

Frames

Registration:

- Registers from database frame elements using Locale Filters
- CoCreate Model Manager Java code builds Lisp registration code for frames

Java Code:

The code is located in:

com.osm.datamgmt.biz.DefaultAnnotationFrameRegister

```
private void writeFileAnnotation(Frame2D frame, PrintWriter out)
throws WMEException {
    String frameName = frame.getName();
    out.println("(docu::docu-register-frame :ui-name \"WM-\" +
frameName + \"\":wm-name \"\" + frameName + \"\")");
}
```

Files:

- Gets files from database frame classes based on CoCreate Model Manager built Lisp registration code

Java Code:

The code is located in:

com.osm.datamgmt.biz.DefaultAnnotationFrameRegister

```
protected void sendFileToAnnotation(File f)
throws WMEException {
    DataManager.getInstance().getAnnotation().initTitleBlocks(f);
}
```

BOM Table and Position Number Templates

Non-Unicode Databases: (Oracle 9i)

Registration List:

- The Lisp code for registering BOM table templates and Position Numbers is generated in CoCreate Model Manager Java code.
- The list of registered entries is located in:

com.osm.datamgmt.biz.DefaultAnnotationBomFlagRegister

Files:

- Gets files from a Locale directory in the CoCreate Modeling installation directories. The Locale directory is chosen by Locale of Schema.
- Example Locale directories:
Modeling_Dir\personality\sd_customize\ANNOTATION
or
Modeling_Dir\personality\<locale_dir>\sd_customize\ANNOTATION

Unicode Databases: (Microsoft SQL Server, Oracle Unicode Databases)

Registration List:

- The Lisp code for registering BOM table templates and Position Numbers is generated in CoCreate Model Manager Java code.
- The list of registered entries is located in:
`com.osm.datamgmt.biz.DefaultAnnotationBomFlagRegister`

Files:

- Gets files from a Locale directory in the CoCreate Modeling installation directories. The Locale directory is chosen by the first BOM filter that defines a locale.
- Example Locale directories:
`Modeling_Dir\personality\lsd_customize\ANNOTATION`
or
`Modeling_Dir\personality\<locale_dir>\lsd_customize\ANNOTATION`
- Unicode Databases use the BOM files whose names end with “_sql”.

CoCreate Drafting

Frames

Registration:

- Registers from database using Locale Filters.
- Java code creates code that builds toolbar macro.

Java Code:

```
com.osm.datamgmt.biz.DefaultDraftingFrameRegister
```

Files:

- Gets files from database based on toolbar macro
- Java Code created code that build toolbar macro

Java Code:

```
com.osm.datamgmt.biz.DefaultDraftingFrameRegister
```

BOM Table and Position Number Templates

The same registration processes are used in CoCreate Drafting for Oracle and SQL Server databases.

Available Names:

- All available names must be defined in the <Mappings> section of the *xml* file. The mappings define which elements are BOM position number templates and which elements are BOM table templates.
- Mapping tags:
 - <BomTable>
 - <BomPositionFlag>

- o Example:

```
<!-- Define all BomPositionFlags in the database. Add a new BomPositionFlag tag for all new BOM
Flags which people should be able to choose -->
<BomPositionFlag catalog="awm_stda" msg_num="123">ISO</BomPositionFlag>
<BomPositionFlag catalog="awm_stda" msg_num="122">DIN</BomPositionFlag>
<BomPositionFlag catalog="awm_stda" msg_num="124">USER</BomPositionFlag>
<!-- Define all BomTables in the database. Add a new BomTable tag for all new BOM tables which
people should be able to choose -->
<BomTable catalog="awm_stda" msg_num="125">A</BomTable>
<BomTable catalog="awm_stda" msg_num="126">B</BomTable>
<BomTable catalog="awm_stda" msg_num="127">C</BomTable>
```

Initial Values:

- o Initial values come from xml option tags:

- <DefaultBomPositionFlagName>
- <DefaultBomTableName>

- o Example:

```
<DefaultBomPositionFlagName>ISO</ DefaultBomPositionFlagName >
<DefaultBomTableName>A</DefaultBomTableName>
```

Preferences List:

- o The Initial values can be changed in the BOM section of the CoCreate Model Manager preferences window.
- o The pull-down list of names in the **Edit > Preferences** window comes from a locale-filtered list of the available names.

Files:

- o Gets files from database based on current preference values.
- o The file is chosen by the local filtered list of available names.

CoCreate Drafting Frame Registration commands in *ui_toolbar.m*

Desktop and Design Management use the frame registration commands in the macro `Toolbar_title_block_load`. This macro is defined in:

```
<CoCreate install directory>\CoCreate Model and Drawing Manager
2008\desktop\aipls\me10\startup\uitoolbar.m
```

(Your directory path may be different based on your installation.)

CoCreate Drawing Manager versions prior to 2006 used the `Toolbar_title_block_load` macro to register frames. CoCreate Drawing Manager 2006 and later only uses the `Toolbar_title_block_load` macro if automatic frame registration is turned off.

CoCreate Annotation Frame Registration commands in *am_wm_customize*

Desktop / Design Data Management commands

The following frame registration section in *am_wm_customize* is active when using Desktop or Design Data Management, but it is **not** active when using CoCreate Model Manager.

Sample location of *am_wm_customize* file:

```
<CoCreate Modeling install directory>\personality
lsd_customize\ANNOTATION\am_wm_customize
```

Frame registration section from *am_wm_customize*:

```
;;-----  
-----  
;; register WM frames to the user interface  
;;-----  
-----  
(when (not (and (functionp 'modelmanager::is-connected)  
(modelmanager::is-connected)))  
  (docu-register-frame :ui-name "WM-A0" :wm-name "A0" )  
  (docu-register-frame :ui-name "WM-A1" :wm-name "A1" )  
  (docu-register-frame :ui-name "WM-A2" :wm-name "A2" )  
  (docu-register-frame :ui-name "WM-A3" :wm-name "A3" )  
  (docu-register-frame :ui-name "WM-A4" :wm-name "A4" )  
  (docu-register-frame :ui-name "WM-A" :wm-name "A" )  
  (docu-register-frame :ui-name "WM-B" :wm-name "B" )  
  (docu-register-frame :ui-name "WM-C" :wm-name "C" )  
  (docu-register-frame :ui-name "WM-D" :wm-name "D" )  
  (docu-register-frame :ui-name "WM-E" :wm-name "E" )
```

Un-registering the default CoCreate Annotation frames

The *am_customize* file registers the frames supplied in the CoCreate Annotation file directories.

```
;;-----  
;; register standard frame sizes for recognition during migration  
(A..E,A0..A4)  
;;-----  
(docu-register-migr-frame-standard-sizes)
```

These frames show up with the following values in the Frame selection pull down list:
A, B, C, D, E, A0, A1, A2, A3, A4, none

You can un-register the default CoCreate Annotation frames with the following command.

```
(docu-unregister-frame-all)
```

The `(docu-unregister-frame-all)` command also un-registers the “none” frame entry.
You can re-register the “none” frame with the following command.

```
(docu:docu-register-frame :ui-name docu::*docu-none-frame* :wm-name "")
```

The key here is the variable `docu::*docu-none-frame*` for the ui-name.

Some installations saw error messages about the “none” frame when using the `(docu-unregister-frame-all)` command. You can avoid this by individually unregistering all of the default CoCreate Annotation frames, but not the “none” frame.

```
(docu-unregister-frame "A" )  
(docu-unregister-frame "B" )  
(docu-unregister-frame "C" )  
(docu-unregister-frame "D" )  
(docu-unregister-frame "E" )  
(docu-unregister-frame "A0" )  
(docu-unregister-frame "A1" )  
(docu-unregister-frame "A2" )  
(docu-unregister-frame "A3" )  
(docu-unregister-frame "A4" )
```

How to turn off the automatic registration processes

CoCreate Annotation Frames:

The **<AnnotationFrameRegister>** xml tag can be used to turn off the automatic frame registration and only use registration commands in the am-customize or am-wm-customize files. The tag is normally used to define the Java class the controls the frame registration. The automatic CoCreate Annotation frame registration process will be disabled if the tag is used without a Java class definition

```
<AnnotationFrameRegister></AnnotationFrameRegister>
```

CoCreate Annotation BOM registration:

Setting the value for the **<EnableAnnotationBomRegister>** xml tag to false will disable the automatic BOM registration processes.

```
<EnableAnnotationBomRegister>>false</EnableAnnotationBomRegister>
```

CoCreate Drafting Frames:

The **<DraftingFrameRegister>** xml tag that can be used to turn off the automatic frame registration and only use registration commands in the regular macro files. The tag is normally used to define the Java class the controls the CoCreate Drafting frame registration. The automatic CoCreate Drafting frame registration process will be disabled if the tag is used without a Java class definition

```
<DraftingFrameRegister></DraftingFrameRegister>
```

CoCreate Drafting BOM registration:

The CoCreate Drafting BOM Flag and BOM Table registration is manually controlled in the xml file by the following flags:

```
<BomPositionFlag>  
<BomTable>  
<DefaultBomPositionFlagName>  
<DefaultBomTableName>
```

See CoCreate Drafting "[BOM Table and Position Number Templates:](#)" section in this article for an explanation of the CoCreate Drafting BOM registration process.

Customizing CoCreate Annotation BOM registration:

The BOM Table and Position Number file registration is coded in `com.osm.datamgmt.biz.DefaultAnnotationBomFlagRegister`

To add or change the registration process, you could extend the Java code, or it may be simpler to modify the Lisp registration commands used by Desktop and Design Management in the *am-wm-customize* file.

Locale Filters

In CoCreate Manager Server 2006 the SQL Server database was added to the list of supported databases. Unlike Oracle 9i, SQL Server databases are not language specific. Since the database is not language specific, frames and bom_flag elements are stored for all six of the supported languages. SQL Server databases can store characters for all languages because it uses Unicode.

(In CoCreate Manager Server 2006 Oracle 9i databases, only the frame and bom_flag elements that match the language of your database are stored in the database.)

Starting with CoCreate Manager Server 2006, CoCreate Model Manager automatically registers frames from the database into CoCreate Annotation. In order to prevent all the frames from all the languages from being registered, some filtering routines were added that limit the frame registration. The FRAME_2D and BOM_FLAG classes have a new attribute called LOCALE. This attribute is used to determine the language of the element. The default LOCALE values are en, fr, de, it, ja, and es.

Default Filter Definitions

CoCreate Manager Server 2006

The following filter definitions are from the CoCreate Manager Server 2006 *wmconf.xml* file. If you have elements with the same name but different languages, only the English element will be allowed through. The filter will however let all elements through if there is only one with a given name.

```
<FrameFilters>
  <LocaleFilter>
    <FilterClass>
      com.osm.biz.util.FilterMultiLocaleKeepSpecifiedLocale
    </FilterClass>
    <Index>1</Index>
    <Parameters>
      <Locale>en</Locale>          <<< Defaults to English
    </Parameters>
  </LocaleFilter>
</FrameFilters>
```

```
<BomDrawingFilters>
  <LocaleFilter>
    <FilterClass>
      com.osm.biz.util.FilterMultiLocaleKeepSpecifiedLocale
    </FilterClass>
    <Index>1</Index>
    <Parameters>
      <Locale>en</Locale>          <<< Defaults to English
    </Parameters>
  </LocaleFilter>
</BomDrawingFilters>
```

Example Filter Results:

By default, there are six versions of the following frames in a 14.00 Unicode database. A, B, C, D, E, A0, A1, A2, A3, A4. Each of the frame names has a version for each of the six default languages making a total of sixty frames stored in the database.

For this example we will say that the database administrator also stored 2 new frames named "big_frame" and "small_frame" using the new save frame option in CoCreate Model Manager.

The *wmconf.xml* file filters will remove all but the ten English default frames from the registration list. There is only one version of "big_frame" and "small_frame", so these frames will also be added to the registration list.

Version 14.20 of CoCreate Manager Server 2006 and later

The *wmconf.xml* file contains two filters to control frame and bom_flag registration:

- FilterMultiLocaleKeepCurrent
- FilterMultiLocaleKeepEnglish

```
<FrameFilters>
  <LocaleFilter>
    <FilterClass>
      com.osm.biz.util.FilterMultiLocaleKeepCurrent
    </FilterClass>
    <Index>1</Index>
  </LocaleFilter>
  <LocaleFilter>
    <FilterClass>
      com.osm.biz.util.FilterMultiLocaleKeepEnglish
    </FilterClass>
    <Index>2</Index>
  </LocaleFilter>
</FrameFilters>
```

```
<BomDrawingFilters>
  <LocaleFilter>
    <FilterClass>
      com.osm.biz.util.FilterMultiLocaleKeepCurrent
    </FilterClass>
    <Index>1</Index>
  </LocaleFilter>
  <LocaleFilter>
    <FilterClass>
      com.osm.biz.util.FilterMultiLocaleKeepEnglish
    </FilterClass>
    <Index>2</Index>
  </LocaleFilter>
</BomDrawingFilters>
```

Filter Index 1

FilterMultiLocalKeepCurrent extends FilterMultiLocaleKeepOneLocale
Filter method from FilterMultiLocaleKeepOneLocale

Filter Results:

- If only one element of a given name is found, then keep it.
- If more than one element of a given name is found, and the locale matches the client locale, then keep it. Drop others with that name.
- If more than one element of a given name is found, and none matches the client locale, then keep them all.

Filter Index 2

FilterMultiLocalKeepEnglish extends FilterMultiLocaleKeepOneLocale

Receives from Filter 1:

- Gets single name elements.
- Gets elements matching client locale
- Gets multiple elements with same name but none matched client locale

Filter results:

- If only one element of a given name is found, then keep it.
- If more than one element of a given name is found, and the locale matches English locale, then keep it. Drop others with that name.
- If more than one element of a given name is found, and none matches English locale, then keep them all.

How to override *wmconf.xml* default filters

Use the `<clear/>` xml tag and create a new set of filter tags

```
<FrameFilters>
  <clear/>      <<<< The clear tag deletes existing FrameFilter definitions.
  <LocaleFilter>
    <FilterClass>
      com.osm.biz.util.FilterMultiLocaleKeepSpecifiedLocale
    </FilterClass>
    <Index>1</Index>
    <Parameters>
      <Locale>de</Locale>  <<<< Locale changed from en to de.
    </Parameters>
  </LocaleFilter>
</FrameFilters>
```

Sample Name Restriction Filter Definition

Additional frame filters can be defined in customized *xml* files. The following definition will limit frame registration based in the frame names.

The client for the following example is running in English. The database administrator also stored two custom frames called "big_frame" and "small_frame" into the database.

In an English locale client, the default frame filter in *wmconf.xml* will remove all but the English frames, allowing the following frames to pass: A, B, C, D, E, A0, A1, A2, A3, A4, big_frame, and small_frame.

This example restricts the list of frames even further so only A, B, C, D, and big_frame are allowed through. (Note that the index in this example is set to "three", so it will not override the filters in the *wmconf.xml* file.)

Since "big_frame" frame is in the name list, it will be registered. "small_frame" is not in the name list, so it will not be registered.

```

<FrameFilters>
  <LocaleFilter>
    <FilterClass>com.osm.biz.util.FilterKeepSpecifiedName</FilterClass>
    <Index>3</Index>
    <Parameters>
      <Name>big_frame</Name>
      <Name>A</Name>
      <Name>B</Name>
      <Name>C</Name>
      <Name>D</Name>
    </Parameters>
  </LocaleFilter>
</FrameFilters>

```

Restricting frame and bom_flag element registration by controlling database elements

Instead of creating custom filters, you can also control frame registrations by adding and deleting frames and bom_flag elements in the database.

Writing Custom Java Filter Classes

There may be some situations where you will need more capabilities than are provided with the filter *xm* tags and the out of the box Java filter classes. Perhaps you would want one set of frames for one set of people, but another set of frames for another set of people. You could write a new Java filter class that would handle different frames for different groups of people. You would access these filters by using the **<FilterClass>** tag in your xml files.

Database Schema Changes for the new LOCALE code

In new CoCreate Manager Server 2006 databases:

- Added a new attribute called LOCALE to the FRAME_2D class
- Added the LOCALE attribute to the unique id of the FRAME_2D class
- Set the LOCALE value for each of the frames stored in the database.
- Only For Unicode Databases (for Version 14.00 that means SQL Server databases):
 - Stored all versions of the frames for all six supported languages in the FRAME_2D class

For Databases that were migrated from an older version to CoCreate Manager Server 2006:

- Added a new attribute called LOCALE to the FRAME_2D class
- Did not change the unique id attribute list for the FRAME_2D class
- Did not store additional language versions of the frames in the database
- Did not set the LOCALE value for any existing frames

In 14.00 the LOCALE field is not given a value when saving new FRAME_2D or BOM_FLAG elements.

In 14.20 the LOCALE field is set to the current client LOCALE value when saving new FRAME_2D and BOM_FLAG elements to the database.

How to find more information about locale filters

The XML section of the *CoCreate Model Manager Customization Guide* has more information about locale filters.

Java filter classes

The individual Java filter classes are not documented in the customization guide. You will need to read the Java code or JavaDoc to see which filters are available and how each filter works.

Java Locale Filters are located in *com.osm.biz.util*

The following filter classes are defined in the 14.00 Java code.:

FilterKeepSpecifiedName	Filters out all but specified names
FilterMultiLocaleKeepCurrent	Keeps elements with current JVM locale
FilterMultiLocaleKeepEnglish	Keeps elements with English locale
FilterMultiLocaleKeepOneLocale	Keeps elements with specified locale
FilterMultiLocaleKeepSpecifiedLocale	Used in Version 14.00 only

The filtering process is controlled by locator classes in *com.osm.datamgmt.biz*

Frame filtering is controlled by the *FrameLocator* class.

BOM Flag filtering is controlled by the *BomFlagLocator* class.

The classes *FrameLocator* and *BomFlagLocator* extend *LocaleDependentLocator*.

The *getFilteredElements* method defined in *LocaleDependentLocator* gathers the list of elements, and the list of filters. The method will then loop through each of the filters, and filter the element list.

```
public List getFilteredElements() throws WMEException {
    List allElements = getAllElements();
    List filters = getFilters();
    List filteredElements = allElements;

    for (int i = 0; i < filters.size(); i++) {
        ILocaleFilter filter = (ILocaleFilter) filters.get(i);
        filteredElements = filter.filter(filteredElements);
    }

    return filteredElements;
}
```

Sample Filter from a Java Filter Class:

com.osm.biz.util.FilterMultiLocaleKeepOneLocale

```
public List filter(List localizedObjects) {
    if ((desiredLocale == null) && (localizedObjects != null)) {
        return new ArrayList(localizedObjects);
    }

    List keptObjects = new ArrayList();

    if (localizedObjects == null) {
        return keptObjects;
    }

    HashMap nameMap = createNameMap(localizedObjects);

    Iterator it = nameMap.keySet().iterator(); <<<< find all elements with
same name

    while (it.hasNext()) {
        String name = (String) it.next();
        List mapList = (List) nameMap.get(name);

        if (mapList.size() == 1) {
            keptObjects.add(mapList.get(0)); <<<<pass element if there
<<<<is only one element
<<<<with that name.
        } else {

            boolean hasSysLocale = false;

            for (int i = 0; i < mapList.size(); i++) {<<<<filter out extra local
entries
                ILocalizedObject currObject = (ILocalizedObject) mapList.get(i);

                if ((currObject.getLocale() == null) ||
(currObject.getLocale().length() == 0)) {
                    continue;
                }

                if
(desiredLocale.toLowerCase().startsWith(currObject.getLocale().toLowerCase()))
{
                    hasSysLocale = true;
                    keptObjects.add(currObject);
                }

                if (!hasSysLocale) {
                    for (int i = 0; i < mapList.size(); i++) {
                        keptObjects.add(mapList.get(i));
                    }
                }
            }
        }

        return keptObjects; <<<< Pass filtered list back to program
    }
}
```

XML Tags related to Registration Processes and Locale Filters

The following xml tags are related to registration processes and locale filters. These tags are documented in the XML section of the CoCreate Model Manager Customization Guide.

```
<AnnotationFrameRegister>  
<BomDrawingFilters>  
<BomPositionFlag>  
<BomTable>  
<clear/>  
<DefaultAnnotationBomFlagRegister>  
<DefaultBomPositionFlagName>  
<DefaultBomTableName>  
<DraftingFrameRegister>  
<EnableAnnotationBomRegister>  
<FilterClass>  
<FrameFilters>  
<Index>  
<Locale>  
<LocaleFilter>  
<Parameters>
```

Step by Step Instructions

How to add custom frames to the database

This section describes the process for creating and saving new title blocks in CoCreate Model Manager. Title blocks can be created in CoCreate Annotation or CoCreate Drafting. If you are using CoCreate Drafting and CoCreate Annotation, it is recommended that title blocks be created in CoCreate Drafting. If you do not use CoCreate Drafting, then it is easier to create the title blocks in CoCreate Annotation.

To create and save title blocks in CoCreate Annotation

Note: CoCreate Annotation frames display an associativity warning when used in CoCreate Drafting.

Steps:

1. Start CoCreate Modeling and CoCreate Annotation.
2. Create or load a model.
3. Create a drawing in CoCreate Annotation. Select a **WM_<frame>** from the *Sheet Frame* dropdown list.
4. Save the drawing.
5. Select **Text > Reference > Assign**. Change a text reference.
6. Select **Data Management > Save Frame** (from the Database icon).
7. Select **New Frame** from the *Database Frame* dropdown list. Enter a name in the *DB Name* field.
8. To make sure the new frame is available, select **Drawing > Modify Frame > Type**. The new frame should show up in the **New Frame** dropdown list.
9. If you are using CoCreate Drafting and CoCreate Drawing Manager, the new title block will show up in the **Title-Block > Load** list.

To create and save title blocks in CoCreate Drafting

Initial steps:

1. Start CoCreate Drafting and CoCreate Drawing Manager.
2. Select a title block in the **Title-Block > Load** list.
3. Modify a text reference in this title block. (**TB-Config > Assign References**, select from TEXT REF NAMES, click text to change).
4. Use the CoCreate Drafting menu, **Parts > Part Browser**, and then edit the `.tb_frame_name` to change the "name" portion to a value of your choosing. This "name" will be saved in the DRAWING_2D FORMAT attribute. Do not change the `.tb_frame_` portion.
5. Save this locally as an **mi** file.
6. If only CoCreate Model Manager is being used, and Design Management or Desktop will never be used, complete the steps in the **CoCreate Model Manager only** section. Otherwise, complete the steps in the **CoCreate Model Manager, Design Management or Desktop** section. This is necessary because the **CoCreate Model Manager only** method does not save the file type, which Design Management and Desktop must have to find title blocks.

CoCreate Model Manager only

This imports the file using CoCreate Drawing Manager

Steps:

1. Edit your custom *xml* to allow FRAME_2D items to be created in CoCreate Drawing Manager by adding the following:

```
<Class extends="BASE_FRAME_2D">
  <Name catalog="awm_stda" msg_num="256">FRAME_2D</Name>
  <CanCreateInUI>true</CanCreateInUI>
</Class>
```

2. Restart CoCreate Drawing Manager.
3. Select **File > New > Document**.
4. Select **FRAME_2D** from the *Class* dropdown list.
5. Right click on the new FRAME_2D item in the workspace and select **Import**.
6. Browse to the file that was saved in step 4 of the **Initial Steps** section and select it.
7. Restart CoCreate Drawing Manager.
8. Select the new title block in the **Title-Block > Load** menu.

CoCreate Model Manager, Design Management or Desktop

This creates a frame element and imports the file using the CoCreate Manager Server Client.

1. Start the CoCreate Manager Server Client.
2. Query for My Packets (**Query > Show > My Packets**).
3. Double click on the packet.
4. Select **Edit > Create > Document**.
5. Select **FRAME_2D** from the *CLASS* dropdown list.
6. Enter a name in the *CREATE* dialog.
7. Select **File > Import** from the new frame document dialog.
8. In the *IMPORT* dialog:
 - a. Select the directory that the file was saved to in step 4 of the **Initial Steps** section.
 - b. In the *File Name* field, enter the file name of the file saved in step 4 of the **Initial Steps** section.
 - c. Enter **Model.MI** in the *File type* field.
9. Restart CoCreate Drawing Manager.
10. Select the new title block in the **Title-Block > Load** menu.

How to add custom CoCreate Drafting BOM templates to the Database

You can create a new BOM template in CoCreate Drafting, or modify an existing template from the database.

Load a BOM table template or Position number template from the database

The BOM table templates and Position number templates are stored in the following classes:

Design Management Schema:
BOM_FLAG
Desktop Schema:
me_bom_flag

The BOM table templates are stored with the names A, B, C
The Position number templates are stored with the names DIN, ISO, USER, etc

Using the advanced search option in CoCreate Model Manager, search for elements from the BOM flag class for your schema.

(You may want to use the Workspace layout manager to turn on the "Locale" attribute so you can choose the correct language for the template.)

Select a template, and right click it to bring up the load option. Load the template to CoCreate Drafting.

Modify the Template

Modify the template geometry, text references, and position number connection points.

Save the modified template to the database:

Using the CoCreate Drawing Manager menu in CoCreate Drafting, select **BOM-Config > Store > Pos-No** or **BOM Layout**.

Saving Position Number templates:

If you chose **Pos-No**, a save window appears.
Enter a name and select save.

Saving BOM table templates:

If you chose **BOM Layout**, CoCreate Drafting prompts you to "Identify part for BOM header".

Select the BOM header.

CoCreate Drafting then prompts you to "Identify part for BOM component.

Select the BOM component.

A save window appears.

Enter a name and select save.

Basic Steps for Controlling CoCreate Annotation Frame Registration

Step One: Turn off the default CoCreate Annotation frame registration (all except for the "none" option):

On each computer where you are running CoCreate Modeling: Edit the CoCreate Annotation *am_wm_customize* file and add the following lines to the file:

```
(docu-unregister-frame "A" )  
(docu-unregister-frame "B" )  
(docu-unregister-frame "C" )  
(docu-unregister-frame "D" )  
(docu-unregister-frame "E" )  
(docu-unregister-frame "A0" )  
(docu-unregister-frame "A1" )  
(docu-unregister-frame "A2" )  
(docu-unregister-frame "A3" )  
(docu-unregister-frame "A4" )
```

Sample location for the *am_wm_customize* file.

```
<CoCreate Modeling install directory>\personality  
  \sd_customize\ANNOTATION\am_wm_customize
```

Step Two: Limit the database frame registration to specific database frames:

Option One: Delete unwanted frames from the database.

By default, only users in the sysadmin group can delete frames from the database.

Option Two: Use an XML frame name restriction filter to limit the frame registration:

Add a customized frame filter to the xml file. Only the frames listed by the <Name> tags will be registered. All other database frames will be ignored. The frame filter may be placed anywhere in the file within <WMConfig>, but not within any of the subsections (such as <Options> or <ClassDefs>).

```
<FrameFilters>  
  <LocaleFilter>  
    <FilterClass>com.osm.biz.util.FilterKeepSpecifiedName</FilterClass>  
    <Index>3</Index>  
    <Parameters>  
      <!--Name equals the name of the element in the database -->  
      <Name>Custom_A</Name>  
      <Name>Custom_B</Name>  
      <Name>Custom_C</Name>  
      <Name>Custom_D</Name>  
    </Parameters>  
  </LocaleFilter>  
</FrameFilters>
```

Option Three: Turn off automatic registration of database frames and manually register them in the *am_wm_customize* file:

Step One:

Turn off automatic registration of database frames

Add the following line to the <Options> section of your custom XML file.

```
<AnnotationFrameRegister></AnnotationFrameRegister>
```

Step Two:

On each CoCreate Modeling machine, manually register the custom database frames in the *am_wm_customize* file:

```
(docu-register-frame :ui-name "WM-Custom_A" :wm-name "Custom_A" )  
(docu-register-frame :ui-name "WM-Custom_B" :wm-name "Custom_B" )  
(docu-register-frame :ui-name "WM-Custom_C" :wm-name "Custom_C" )  
(docu-register-frame :ui-name "WM-Custom_D" :wm-name "Custom_D" )
```

Step Three: Test and distribute the modified CoCreate Model Manager XML file.

1. Test your modifications by placing the XML file on a single machine.
 - a. Add the modified XML file to your Application Data directory.

- b. Test the XML file changes by running CoCreate Model Manager.
2. Distribute the new xml file to all users.
- a. Copy the tested XML file to the CoCreate Manager Server “config” directory on the machine where you installed your Software Distribution Server.
 - b. Version 14.00 and Lower:
Run the *mmbuild.bat* file to create a new installation package for the Software Distribution server. The *mmbuild.bat* file is located in the *clntwin* directory. The next time people start CoCreate Model Manager, they will be prompted to re-install CoCreate Model Manager and get the new XML file.

Version 14.20 and Higher:

If you are running CoCreate Model Manager version 14.20 or higher you can skip step B. The xml file will be copied directly from the *config* directory to the user's documents and setting directory each time a client is started. Users will not be prompted to re-install CoCreate Model Manager. They will automatically have the new version of the xml file available to them

NOTE: Rather than editing the *am_wm_customize* file directly in the CoCreate Modeling installation directory, you may prefer to place a modified *am_wm_customize* file in one of the CoCreate Modeling customization directories.

Example:

A file called *am_wm_customize* placed in *C:\Documents and Settings\ will be executed in CoCreate Annotation if you are using CoCreate Model Manager. You may prefer to use the other CoCreate Modeling customization processes like using the “SITE” or “CORP” system variables.*

Basic Steps for Controlling CoCreate Drafting Frame Registration

Step One: Limit the database frame registration to specific database frames:

Option One: Delete unwanted frames from the database.

By default, only users in the sysadmin group can delete frames from the database.

Option Two: Use an XML frame name restriction filter to limit the frame registration:

Add a customized frame filter the xml file. Only the frames listed by the <Name> tags will be registered. All other database frames will be ignored. The frame filter may be placed anywhere in the file within <WMConfig>, but not within any of the subsections (such as <Options> or <ClassDefs>).

```
<FrameFilters>
  <LocaleFilter>
    <FilterClass>com.osm.biz.util.FilterKeepSpecifiedName</FilterClass>
    <Index>3</Index>
    <Parameters>
      <Name>Custom_A</Name>    <<< The name of the element in the
database
      <Name>Custom_B</Name>
```

```

    <Name>Custom_C</Name>
    <Name>Custom_D</Name>
  </Parameters>
</LocaleFilter>
</FrameFilters>

```

Option Three: Turn off automatic registration of database frames and manually register them in the *uitoolbar.m* file:

Step One:

- a. Make a copy of your custom *xml* file in your *Documents and Settings* directory and give it a new name
- b. Turn off automatic registration of database frames.
Add the following line to the <Options> section of the test version of the CoCreate Model Manager custom XML file.

```
<DraftingFrameRegister></DraftingFrameRegister>
```

Step Two

- a. Copy the *Toolbar_title_block_load* macro from *uitoolbar.m* to a file called *ui_cust.m* in your *CoCreate Model Manager and Drawing Manager 2007\desktop\lips\me 10\custom* directory.
- b. Customize the frame registrations in the *Toolbar_title_block_load* macro in the *ui_cust.m* file:

Current *Uitoolbar.m* registration command for frame 'A0':

```

TOOLBAR_ITEM OVER_WRITE "btn_DMLoadA0" PERSISTENT
OFF CAPTION Awm_ttext_2 ITEM_TYPE PUSH_BUTTON
ON_ACTION "Delete_rc_toolbar Create_toolbar_options 'Absolute'
Awmc_m_load_tb_frame 'A0' "" ADD_TO_TOOLBAR 11 END

```

Example of how to add a new frame registration that will show up in the load frame menu after frame 'A0':

```

TOOLBAR_ITEM OVER_WRITE "btn_DMLoadA0" PERSISTENT
OFF CAPTION 'menu_frame_name' ITEM_TYPE PUSH_BUTTON
ON_ACTION "Delete_rc_toolbar Create_toolbar_options 'Absolute'
Awmc_m_load_tb_frame 'db_frame_name' "" ADD_TO_TOOLBAR 12
END

```

'menu_frame_name' This argument controls how the frame will show up in the load frame menu. Note that the existing frame definitions use a macro variable like *Awm_ttext_2*. This example uses text enclosed in quotes.

'db_frame_name' This argument is for the name of the frame as it is stored in the database.

12 This argument defines the line number where the frame will show up in the load frame menu.

Step Two: Test and distribute the frame customization files.

1. Test your modifications on your local machine by running CoCreate Drawing Manager.
2. Distribute the new *xml* file and customized *Toolbar_title_block_load* macro to all users.
 - a. Copy your test *xml* file to the CoCreate Manager Server *config* directory on the machine where you installed your Software Distribution Server.
 - b. Copy the *ui_cust.m* file from your test machine to the *aips\me10\custom* directory on the machine where you installed your Software Distribution Server.
 - c. Run *mmbuild.bat* to create a new installation package for the Software Distribution server. The next time people start CoCreate Model Manager, they will be prompted to re-install CoCreate Model Manager. The re-install process will give the users the new *xml* file and the modified *uitoolbar.m* file.

Basic Steps for Controlling CoCreate Annotation BOM Template Registration

Step One: Turn off the default CoCreate Annotation BOM template registration:

CoCreate Annotation automatically registers a standard set of internal CoCreate Annotation BOM templates. These templates do not use database attribute text references. The default CoCreate Manager Server templates are registered with the DM- prefix in their names. CoCreate Annotation template names do not use the DM- prefix. In order to avoid confusion between the CoCreate Manager Server templates and the CoCreate Annotation standard templates, you may want to un-register the standard CoCreate Annotation templates.

On each CoCreate Modeling machine: Edit the *am_wm_customize* file and add the following line to the file:

```
(DOCU-UNLOAD-ALL-BOM-LAYOUTS)
```

Step Two: Limit the automatic registration to specific BOM Template files:

Option One: Extend Java code

The list of automatically registered CoCreate Annotation BOM table templates and Position Numbers is generated in CoCreate Model Manager Java code. The Java code generates Lisp code in a file that is run in CoCreate Annotation by CoCreate Model Manager. The list of registered templates is in:

```
com.osm.datamgmt.biz.DefaultAnnotationBomFlagRegister
```

Option Two: Turn off automatic registration of CoCreate Model Manager BOM templates and manually register them in the *am_wm_customize* file:

Step One:

Turn off automatic registration of CoCreate Model Manager BOM templates. Setting the value for the **<EnableAnnotationBomRegister>** xml tag to false disables the automatic BOM registration processes. Add the following line to the <Options> section of the custom *xml* file:

```
<EnableAnnotationBomRegister>>false</EnableAnnotationBomRegister>
```

Step Two:

On each CoCreate Modeling machine, manually register the custom CoCreate Model Manager BOM templates, and set the default layouts in the *am_wm_customize* file.

(If you unregistered the standard CoCreate Annotation templates with the (DOCU-UNLOAD-ALL-BOM-LAYOUTS) command, be sure to add the custom layout registration commands after the (DOCU-UNLOAD-ALL-BOM-LAYOUTS) command.)

Example Layout registration commands from *am_wm_customize*:

```
(DOCU-LOAD-BOM-HEAD-LAYOUT :UI-NAME "DM-Custom" :FILE (search-file "sd_customize/ANNOTATION/dmbomh_custom.mi"))
(DOCU-LOAD-BOM-COMP-LAYOUT :UI-NAME "DM-Custom" :FILE (search-file "sd_customize/ANNOTATION/dmbomc_custom.mi"))
(DOCU-LOAD-BOM-FLAG-LAYOUT :UI-NAME "DM-Custom" :FILE (search-file "sd_customize/ANNOTATION/dmbomf_custom.mi"))

;; Set default BOM Layouts
(docu::docu-set-current-bom-flag-layout " DM-Custom ")
(docu::docu-set-current-bom-head-layout " DM-Custom ")
(docu::docu-set-current-bom-comp-layout " DM-Custom ")
(docu::docu-set-current-bom-sketch-layout " DM-Custom ")
```

Note: There is a section in the *am_wm_customize* file that registers BOM layouts for the Desktop and Design Management products. This section of code is **not** executed when running CoCreate Model Manager. Be sure to add your custom CoCreate Model Manager BOM layout registration commands outside of the Desktop-Design Management BOM layout section.

Step Three: Test and distribute the modified CoCreate Model Manager XML file.

1. Test your modifications by placing the XML file on a single machine.
 - a. Add the modified *xml* file to your *Documents and Settings* directory.
 - b. Test the *xml* file changes by running CoCreate Model Manager.

- 2: Distribute the new *xml* file to all users.
 - a. Copy the tested *xml* file to the CoCreate Manager Server *config* directory on the machine where you installed your Software Distribution Server.
 - b. Version 14.00 and Lower:

Run *mmbuild.bat* to create a new installation package for the Software Distribution server. The *mmbuild.bat* file is located in the *clntwin* directory. The next time people start CoCreate Model Manager, they will be prompted to re-install CoCreate Model Manager and get the new *xml* file.

Version 14.20 and Higher:

If you are running CoCreate Model Manager version 14.20 or higher you can skip step B. The *xml* file will be copied directly from the *config* directory to the user's documents and setting directory each time a client is started. Users will not be prompted to re-install CoCreate Model Manager. They will automatically have the new version of the *xml* file available to them

NOTE: Rather than editing the *am_wm_customize* file directly in the CoCreate Modeling installation directory, you may prefer to place a modified *am_wm_customize* file in one of the CoCreate Modeling customization directories.

Example:

A file called *am_wm_customize* placed in *C:\Documents and Settings\<user login>\Application Data\CoCreate\OSD_Modeling\<version>\Annotation* will be executed in CoCreate Annotation if you are using CoCreate Model Manager. You may prefer to use the other CoCreate Modeling customization processes like using the "SITE" or "CORP" system variables.

Basic Steps for Controlling CoCreate Drafting BOM Template Registration

Step One: Register the Templates in the *xml* file:

Register the BOM table templates and Position number templates in the <Mappings> section of the *xml* file using the <BomPositionFlag > and the <BomTable> tags. Registering the templates will make them available in the Workspace BOM tab of the Preferences window.

Example:

```
<Mappings>
<!-- Define all BomPositionFlags in the database. Add a new BomPositionFlag
tag for all new BOM Flags which people should be able to choose -->
<BomPositionFlag catalog="awm_stda" msg_num="123">
  ISO
</BomPositionFlag>
<BomPositionFlag catalog="awm_stda" msg_num="122">
  DIN
</BomPositionFlag>
<BomPositionFlag catalog="awm_stda" msg_num="124">
  USER
</BomPositionFlag>
<BomPositionFlag>CUSTOM_FLAG</BomPositionFlag>
<!-- Define all BomTables in the database. Add a new BomTable tag for all
new BOM tables which people should be able to choose -->
<BomTable catalog="awm_stda" msg_num="125">A</BomTable>
<BomTable catalog="awm_stda" msg_num="126">B</BomTable>
<BomTable catalog="awm_stda" msg_num="127">C</BomTable>
```



```
<BomTable>C_CUSTOM_TABLE</BomTable>
</Mappings>
```

Remove any entries from the lists that you do not want to use.

Step Two: Define the names of the default templates.

CoCreate Drawing Manager uses the templates that are defined in the BOM tab of the Workspace Preferences window. You can change the settings in the BOM preferences window, but the initial values are defined in the <Options> sections of the *xml* file using the <DefaultBomPositionFlagName> and <DefaultBomTableName> tags:

Example:

```
<Options>
<DefaultBomPositionFlagName>CUSTOM_FLAG
</DefaultBomPositionFlagName>
<DefaultBomTableName>C_CUSTOM_TABLE</DefaultBomTableName>
</Options>
```

Step Three: Test and distribute the modified CoCreate Model Manager *xml* file.

1. Test your modifications by placing the XML file on a single machine.
 - a. Add the modified *xml* file to your *Documents and Settings* directory.
 - b. Test the *xml* file changes by running CoCreate Model Manager.

- 2: Distribute the new *xml* file to all users.
 - a. Copy the tested *xml* file to the CoCreate Manager Server *config* directory on the machine where you installed your Software Distribution Server.
 - b. Version 14.00 and Lower:
Run *mmbuild.bat* to create a new installation package for the Software Distribution server. The *mmbuild.bat* file is located in the *clntwin* directory. The next time people start CoCreate Model Manager, they will be prompted to re-install CoCreate Model Manager and get the new *xml* file.

Version 14.20 and Higher:
If you are running CoCreate Model Manager version 14.20 or higher you can skip step B. The *xml* file will be copied directly from the *config* directory to the user's documents and setting directory each time a client is started. Users will not be prompted to re-install CoCreate Model Manager. They will automatically have the new version of the *xml* file available to them

How to control if frames are loaded or saved with drawings:

Frame Saving Process:

By default, Model Manager saves the frames with an Annotation drawing.
By default, Model Manager does not save the frame with a Drafting drawing.

Frame Loading Process:

As of Model Manager 13.20, when loading a drawing to either Drafting or Annotation, by default, the program will automatically check if a frame needs to be loaded. If the program detects that a frame exists on a drawing, the program will not attempt to load a new frame. If the program detects that the drawing does not have any frames, the program will load new frames.

How to control if frames are loaded or saved with drawings:

There are several flags that can be used to modify the default behavior of how frames are loaded and saved with drawings:

Awmc_val_store_with_tb	Used in Drafting and Annotation
Awmc_val_tbname_prefix	Used in Drafting and Annotation
Awmc_val_auto_relod_of_2d_frames	Used in Drafting
Awmc_val_never_reload_2d_frames	Used in Drafting
db-auto-relod-of-2d-frames	Used in Annotation
db-never-reload-2d-frames	Used in Annotation

Which variables are important for the frame handling?

Drafting

Variable name	Values	Description	Where to set?
Awmc_val_store_with_tb	TRUE → store with tb FALSE → store without tb e.g. LET Awmc_val_store_with_tb (TRUE)	defines if the drawing should be stored with or without frame.	%Drafting_DIR%\desktop\aiaps\me10\startup\cust.m
Awmc_val_tname_prefix	.tb_frame_ → English frame .sfeld_rahmen_ → German frame .cadre_ct_ → French frame e.g. LET Awmc_val_tname_prefix 'tb_frame_' Also user defined values can be used. For customers with multi sites in different countries, there is the possibility to define alternative prefixes. {* part name prefix for title block parts*} {! LET Awmc_val_tname_prefix 'tb_frame_' !} {* alternative part name prefixes for title block parts to be reloaded to the current title block from the database*} {! LET Awmc_val_tname_alt_prefix_1 'sfeld_rahmen_' !} {! LET Awmc_val_tname_alt_prefix_2 'cadre_ct_' !} {! LET Awmc_val_tname_alt_prefix_3 'tb_frame_' !} {! LET Awmc_val_tname_alt_prefix_4 " !} {! LET Awmc_val_tname_alt_prefix_5 " !}	defines the prefix for the drafting part name of the frame. e.g. for an A3 frame LET Awmc_val_tname_prefix 'tb_frame_' plus the frame's drafting part name results in: 'tb_frame_A3'	%Drafting_DIR%\desktop\aiaps\me10\startup\cust.m
Awmc_val_auto_reload_of_2d_frames	FALSE → do not auto reload frames TRUE → auto reload frames e.g. LET Awmc_val_auto_reload_of_2d_frames (FALSE)	defines if a frame should be reloaded. When you have drawings with embedded frames, the frame should be deleted in the dwg and reloaded	%Drafting_DIR%\desktop\aiaps\me10\startup\cust.m
Awmc_val_never_reload_2d_frames	FALSE → do reload frames if auto-reload is set TRUE → never reload frames e.g. LET Awmc_val_never_reload_2d_frames (FALSE)	even if the variable Awmc_val_auto_reload_of_2d_frames is set to true, do not reload the frame	%Drafting_DIR%\desktop\aiaps\me10\startup\cust.m

	Awmc_val_auto_reload_of_2d_frames ON Awmc_val_never_reload_2d_frames OFF	Awmc_val_auto_reload_of_2d_frames OFF Awmc_val_never_reload_2d_frames OFF	Awmc_val_auto_reload_of_2d_frames ON OFF Awmc_val_never_reload_2d_frames ON
Drawing with embedded Frame	Get frame from DB	Keep frame stored in drawing	Keep frame stored in drawing
Drawing with referenced Frame	Get frame from DB	Get frame from DB	Do not load referenced frame

Annotation

Variable name	Values	Description	Where to set?
Awmc_val_store_with_tb	<p>TRUE → store with tb FALSE → store without tb</p> <p>e.g. LET Awmc_val_store_with_tb (TRUE)</p>	defines if the drawing should be stored with or without frame.	%Annotation_DIR%\win\desktop\aips\me10\startup\cust.m
Awmc_val_tbnam_e_prefix	<p>.tb_frame_ → English frame .sfeld_rahmen_ → German frame .cadre_ct_ → French frame</p> <p>e.g. LET Awmc_val_tbnam_e_prefix '.tb_frame_'</p> <p>Also user defined values can be used.</p> <p>For customers with multi sites in different countries, there is the possibility to define alternative prefixes.</p> <pre>{ * part name prefix for title block parts *} {! LET Awmc_val_tbnam_e_prefix '.tb_frame_' !}</pre> <p>{ * alternative part name prefixes for title block parts to be reloaded to the current title block from the database *}</p> <pre>{! LET Awmc_val_tbnam_e_prefix_1 '.sfeld_rahmen_' !} {! LET Awmc_val_tbnam_e_prefix_2 '.cadre_ct_' !} {! LET Awmc_val_tbnam_e_prefix_3 '.tb_frame_' !} {! LET Awmc_val_tbnam_e_prefix_4 "" !} {! LET Awmc_val_tbnam_e_prefix_5 "" !}</pre>	<p>defines the prefix for the drafting part name of the frame.</p> <p>e.g. for an A3 frame</p> <pre>LET Awmc_val_tbnam_e_prefix '.tb_frame_'</pre> <p>plus the frame's drafting part name results in:</p> <pre>'.tb_frame_A3'</pre>	%Annotation_DIR%\win\desktop\aips\me10\startup\cust.m
db-auto-reload-of-2d-frames	<p>nil → do not auto reload frames t → auto reload frames</p> <p>e.g. (setf modelmanager::*db-auto-reload-of-2d-frames* nil)</p>	defines if a frame should be reloaded. When you have drawings with embedded frames, the frame should be deleted in the dwg and reloaded	<p>%Modeling_DIR%\personality\ <lang>\sd_customize \ ANNOTATION\am_mm_customize</p> <p>Note: In Desktop / Design Data Management am_mm_customize must be replaced by am_wm_customize.</p>

db-never-reload-2d-frames	nil → reload frames if auto-reload is set t → never reload frames e.g. (setf modelmanager:: *db-never-reload-2d-frames* nil)	even if the variable Awmc_val_auto_reload_of_2d_frames is set to true, do not reload the frame	%Modeling_DIR%\personality\ <lang>\sd_customize\ \ANNOTATION\am_m m_customize Note: In Desktop / Design Data Management am_mm_customize must be replaced by am_wm_customize.
----------------------------------	---	--	---

	db-auto-reload-of-2d-frames ON db-never-reload-2d-frames OFF	db-auto-reload-of-2d-frames OFF db-never-reload-2d-frames OFF	db-auto-reload-of-2d-frames ON OFF db-never-reload-2d-frames ON
Drawing with embedded Frame	Get frame from DB	Keep frame stored in drawing	Keep frame stored in drawing
Drawing with referenced Frame	Get frame from DB	Get frame from DB	Do not load referenced frame

Where are the frame flags defined?

Awmc_val_auto_reload_of_2d_frames

```
<CoCreate Drafting install directory>\desktop\startup\wmdt.mac
<CoCreate Modeling install directory>\Annotation\win\desktop\startup\wmdt.mac
```

Awmc_val_never_reload_2d_frames

```
<CoCreate Drafting install directory>\desktop\startup\wmdt.mac
<CoCreate Modeling install directory>\Annotation\win\desktop\startup\wmdt.mac
```

Awmc_val_store_with_tb

CoCreate Manager Server directories

```
<CoCreate Manager Server install directory>\desktop\profile\  
advanced\aiips\me10\startup\profile.m  
designer\aiips\me10\startup\profile.m  
expert\aiips\me10\startup\profile.m  
expert_classic\aiips\me10\startup\profile.m
```

Drafting directories

```
<CoCreate Drafting install directory>\desktop\aipls\me10\startup  
    enable.m  
    startup.m  
    wmdt.mac
```

Modeling directories

```
<CoCreate Modeling install directory>\Annotation\win\desktop\aipls\me10\startup\  
    enable.m  
    startup.m  
    wmdt.mac
```

```
<CoCreate Modeling install directory>\Annotation\win \desktop\profile\  
advanced\aipls\me10\startup\profile.m  
designer\aipls\me10\startup\ profile.m  
expert\aipls\me10\startup\profile.m  
expert_classic\aipls\me10\startup\profile.m
```

db-auto-reload-of-2d-frames

Defined in compiled Annotation code

db-never-reload-2d-frames

Defined in compiled Annotation code

Where to put customized flag settings:

Drafting Macro commands

Location:

```
<CoCreate Drafting install directory>\desktop\aipls\me10\custom\ui_cust.m
```

Example Commands:

```
LET Awmc_val_store_with_tb (FALSE)  
LET Awmc_val_tbname_prefix '.tb_frame_'  
LET Awmc_val_auto_reload_of_2d_frames (FALSE)  
LET Awmc_val_never_reload_2d_frames (TRUE)
```

Annotation Macro commands

Location:

```
<CoCreate Modeling install directory>\Annotation\win\  
desktop\aipls\me10\custom\ui_cust.m
```

Example Commands:

```
LET Awmc_val_store_with_tb (FALSE)
LET Awmc_val_tbname_prefix '.tb_frame_'
```

Annotation lisp commands

Location:

<CoCreate Modeling install directory>
\\personality\sd_customize\ModelManager\sd_customize_2d

Example Commands:

```
(setf modelmanager::*db-never-reload-2d-frames* nil)
(setf modelmanager::*db-auto-reload-of-2d-frames* nil)
```

Both commands are required to not load a frame

Code Examples:

wmdt.mac

Locations for wmdt.mac

<CoCreate Drafting install directory>\desktop\startup\wmdt.mac

<CoCreate Modeling install directory>
\\Annotation\win\desktop\startup\wmdt.mac

Sections from wmdt.mac

```
DEFINE Awmc_val_auto_reload_of_2d_frames
0
END_DEFINE
```

```
DEFINE Awmc_val_never_reload_2d_frames
0
END_DEFINE
```

```
DEFINE Awmc_val_store_with_tb
0
END_DEFINE
```

(The Awmc_val_store_with_tb variable is used in Awm_i_store_quick, but the value is not defined here.)

```
{* Store drawing with/without title block *}
IF (Annotation_active)
  LET Awmc_val_store_with_tb (TRUE)
ELSE
```



```
LET Awmc_val_store_with_tb (FALSE)
END_IF
```

(The Awmc_val_store_with_tb variable is used in Awm_i_store_metop,
but the value is not defined here.)

```
{* Store drawing with/without title block *}
IF (Annotation_active)
  LET Awmc_val_store_with_tb (TRUE)
ELSE
  LET Awmc_val_store_with_tb (FALSE)
END_IF
```

profile.m

Locations for **profile.m**

```
<CoCreate Manager Server install directory>\desktop\profile\  
  advanced\aips\me10\startup\profile.m  
  designer\aips\me10\startup\ profile.m  
  expert\aips\me10\startup\profile.m  
  expert_classic\aips\me10\startup\profile.m
```

```
<CoCreate Drafting install directory>\desktop\profile\  
  advanced\aips\me10\startup\profile.m  
  designer\aips\me10\startup\profile.m  
  expert\aips\me10\startup\profile.m  
  expert_classic\aips\me10\startup\profile.m
```

```
<CoCreate Modeling install directory>\Annotation\win \desktop\profile\  
  advanced\aips\me10\startup\profile.m  
  designer\aips\me10\startup\profile.m  
  expert\aips\me10\startup\profile.m  
  expert_classic\aips\me10\startup\profile.m
```

Section from the Enable_Profile macro in profile.m

```
        { * Store drawing with/without title block * }  
IF (Annotation_active)  
    LET Awmc_val_store_with_tb (TRUE)  
ELSE  
    LET Awmc_val_store_with_tb (FALSE)  
END_IF
```

enable.m

Locations of enable.m

<CoCreate Drafting install directory>\desktop\aips\me10\startup\enable.m

<CoCreate Modeling install directory>\Annotation\win\
desktop\aips\me10\startup\enable.m

Section from enable.m

```
IF (Annotation_active)  
    { * Speichern eines TB in einer SD-Drawing erlauben * }  
    { ***** }  
    LET Awmc_val_store_with_tb TRUE  
END_IF
```

startup.m

Locations of startup.m

<CoCreate Drafting install directory>\desktop\aips\me10\startup\startup.m

<CoCreate Modeling install directory>\Annotation\win\
desktop\aips\me10\startup\startup.m

Section from startup.m

```
{ * Store a TB in a SD Drawing permit * }  
{ ***** * }  
    LET Awmc_val_store_with_tb (TRUE)  
ELSE  
    LET Annotation_active 0  
END_IF
```

wmdt_me.mac

Location of wmdt_me.mac

<CoCreate Manager Server install directory>\desktop\startup\wmdt_me.mac

Section from the Awm i_store_metop macro in wmdt_me.mac

(Awmc_val_store_with_tb is used in this macro, but it is not defined here.)

```
{* Store drawing with/without title block *}
IF (Annotation_active)
  LET Awmc_val_store_with_tb (TRUE)
ELSE
  LET Awmc_val_store_with_tb (FALSE)
END_IF
```

cust.m

Location of cust.m

<CoCreate Drafting install directory>\desktop\startup\cust.m

<CoCreate Modeling install directory>
\Annotation\win\desktop\aiaps\me10\startup \cust.m

Section from cust.m

Note that the flag definitions in cust.m are all commented out by default

```
{* Store drawing with/without title block *}
{! LET Awmc_val_store_with_tb (FALSE) !}

{* Auto reload of Title Blocks when drawing is loaded *}
{! LET Awmc_val_auto_relod_of_2d_frames (TRUE) !}
{! LET Awmc_val_never_reload_2d_frames (FALSE) !}
```