

ORACLE AGILE ENGINEERING DATA MANAGEMENT - MCAD CONNECTOR FOR UNIGRAPHICS NX VERSION 3.9.2.0

RELEASE NOTES

April 23, 2008
Part No. E12430-01

Contents

Supported Unigraphics and Agile e-series Versions.....	2
Supported Unigraphics Versions.....	2
Supported Agile e-series Versions.....	2
Extensions and Issues	2
Related SR's	3
Changes to the Integration's Environment and Unigraphics Startup	3
Unigraphics Startup.....	3
Minor changes and extensions	3
Batch plotting and PQMGR under NX3/NX4/NX5.....	4
Harmonized Menu structure	4
Embedding Manufacturing and Seedpart Extensions	4
Embedding New Create Dialog and Seedparts.....	7

This page is blank

Supported Unigraphics and Agile e-series Versions

Supported Unigraphics Versions

The following Unigraphics Maintenance Releases and Packs are supported with the Unigraphics Integration Version 3.9.2/3.9.2.0.

- ❑ Unigraphics NX: on demand only
- ❑ Unigraphics NX2: on demand only
- ❑ Unigraphics NX3: Maintenance Release 3.0.4 or higher
- ❑ Unigraphics NX4: Maintenance Release 4.0.1 or higher
- ❑ Unigraphics NX5: Maintenance Release 5.0.2 or higher

For more information on installing the above Unigraphics releases refer to the Unigraphics website (www.ugs.com) and the UG documentation.

Supported Agile e-series Versions

The following Agile e-series releases are supported with the Unigraphics Integration Version 3.9.2.0

- ❑ Agile e5 (PLM 5.1 and PLM 5.0.1)
- ❑ Agile e6 (e6.0.1 and e6.0.2)

Fixed Issues and SR's with ECU 3.9.2

Extensions and Issues

- ❑ Detection of Suppressed Components can be enabled by setting EcuSaveSuppressed=1 in tcl/Ecu.ini
- ❑ New Switch in Create Dialog for Model Assignment Control
- ❑ Fixed Recovery for Filenames containing special characters
- ❑ Modified Renaming Mechanism and fixed Renaming Issues during Load.
- ❑ 64bit Support for NX4 and NX5, requires installation of the Microsoft Visual C++ 2005 Redistributable Package (x64) available at Microsoft's download site.
- ❑ Workaround for NX3/NX4/NX5 SDI Plot Bug by setting Environment in NX startup

```
set UGII_SDI_OVERRIDE_HOME=c:\TEMP\nx
set UGII_PLOT_CONVERT_OLD_BEHAVIOR=ON
```

Related SR's

- ❑ FIS-ID 11671 – Customer specific Language Problem after Drawing Creation
- ❑ FIS-ID 11674 – Bookmark Format Support for new NX5 and old NX4/earlier formats
- ❑ FIS-ID 11559 – Detect Suppressed Components

Changes to the Integration's Environment and Unigraphics Startup

Unigraphics Startup

In a Unigraphics session, the startup of the ECU integration and loading the additional menu extensions uses the standard mechanisms and the standard functions of the Unigraphics menu scripts. Thus the paths to the integration binaries and menu scripts resource must be specified by editing the UG environment file `custom_dirs.dat` or respectively the file with customer specific name. If the specified paths are incorrect, the additional menus are not visible.

When installing the integration out-of-the box the file `custom_dirs.dat` in the ECU's ini-directory is used.

When using a customer-specific file, you need to edit the contents of the variable `UGII_CUSTOM_DIRECTORY_FILE` in the file `../com/file ug2.bat`. The referenced customer-specific file must contain the following lines:

```
$ECU_ROOT/application/$ECU_UGV
$ECU_ROOT/ini/${EDB_VER}${ECU_ADMIN}/${ECU_LANG}
$ECU_ROOT/bin/$ECU_MACH/$EDB_VER/$ECU_UGV
$ECU_ROOT
```

When using your own startup scripts check that all variables used by the integration are set correctly.

The content of the variable `USER_STARTUP` must no longer point to the ECU library like in older versions as the library will be loaded automatically when starting UG.

Minor changes and extensions

Several other changes were made:

- ❑ Tcl command `EcuDeleteVar` removed
- ❑ `uc6494` replaced by `UF_DRAW_open_drawing`
- ❑ missing update in `T_DOC_STR.CAX_COM` after renaming the sub component fixed
- ❑ `QueueStopOnLoad` in `EcuBatchQueue.ini` and new tcl command `EcuGetMemoryLoad`

- ❑ plm_open, plm_create, plm_save, plm_saveas for offline UG integration
- ❑ plm_action callback for enhancements
- ❑ slave drawing procedures corrected
- ❑ ecu_lst.dlg and ecu_sel.dlg now found in application subdirectories too
- ❑ fil_path parameter for docfile.par and checkin of parts located outside the cax_temp directory. Enables checkin of BCT parts from any external location.
- ❑ scenario parameter for subtype SP in *.par parameter files (markup of scenario parts)
- ❑ New menu item for delivering scenario parts, related required and/or modified Scripts are EcuPreReturnScenarioToModel.tcl, EcuLoadScenarios.tcl, EcuPostSaveObject.tcl, EcuPostSaveObjectAs.tcl, EcuPostOpenObject.tcl. Menu item in ecu.men can be included like this:

```
BUTTON PLM_RET_SCE
```

LABEL Load Scenarios...

ACTIONS plm_return_scenario

- ❑ Concurrent conflict if a document is revisioned since it was loaded or saved is detected and prevents the part from saving until it is reloaded from PLM (EcuKernel.tcl)
- ❑ Create or Save As into users local directories doesn't allow overwriting of existing files anymore. (EcuKernel.tcl, EcuCreateDialog.tcl, EcuCreateBySeedpart.tcl)
- ❑ Show up a warning message when an assembly is saved on local disc only. (Save option User in save wizard) (EcuKernel.tcl)

Batch plotting and PQMGR under NX3/NX4/NX5

The feature PQMGR from older versions is no longer available in Unigraphics. To get neutral formats or a plot, a file in CGM format needs to be created first which requires additional adaptations for generating additional formats (e.g. HPGL). For further information and requirements contact Oracle.

Harmonized Menu structure

Starting with Unigraphics NX3, only menus in menuscrypt format are supported. The harmonized menu file agile.men is delivered with the integration and can be found in the directory ini/e6/eng/startup/agile.men. For reference the old style menu file is also contained but renamed to ecu.men_.

Each "startup" subdirectory contains the respective file.

Embedding Manufacturing and Seedpart Extensions

Support of Manufacturing requires support of additional seedparts and triggers. Triggers for creation of new Manufacturing parts will create a occurrence to the currently loaded UG model within the new Manufacturing part. Please refer to this table for the required enviroment settings to be made in ini/ecu_ini.bat:

Environment variable containing the full path to the seedpart	Subtype environment (Standard)	Seedpart description
ecu_blank_part		Base Part metric
ecu_blank_part_in		Base Part english
ecu_blank_drw	cax_dp_ident=DP	Base Drawing metric
ecu_blank_drw_in	cax_dp_ident=DP	Base Drawing english
ecu_blank_cam	cax_mp_ident=MP	Base Manufacturing metric
ecu_blank_cam_in	cax_mp_ident=MP	Base Manufacturing english
ecu_blank_sce	cax_sp_ident=SP	Base Scenario metric
ecu_blank_sce_in	cax_sp_ident=SP	Base Scenario english

Refer to this tables, which triggers are called for each subtype and what function number is required in ini/e6/ufcre.men.

Subtype	ufcre.men function number	TCL pre action trigger	TCL post action trigger
Unit mm			
Part	100	EcuPreCreatePart_mm.tcl	EcuPostCreatePart_mm.tcl
Drawing (DP)	116	EcuPreCreateDrawing_mm.tcl	EcuPostCreateDrawing_mm.tcl
Manufacturing (MP)	122	EcuPreCreateCAMPart_mm.tcl	EcuPostCreateCAMPart_mm.tcl
Scenario (SP)	124	EcuPreCreateScenario_mm.tcl	EcuPostCreateScenario_mm.tcl

USER wit UG Standard seedpart	102	none	none
USER with ECU seedpart	121	EcuCreateBySeedPart.tcl	none

Subtype Unit inch	ufcre.men function number	TCL pre action trigger	TCL post action trigger
Part	101	EcuPreCreatePart_inch.tcl	EcuPostCreatePart_inch.tcl
Drawing (DP)	117	EcuPreCreateDrawing_inch.tcl	EcuPostCreateDrawing_inch.tcl
Manufacturing (MP)	123	EcuPreCreateCAMPart_inch.tcl	EcuPostCreateCAMPart_inch.tcl
Scenario (SP)	125	EcuPreCreateScenario_inch.tcl	EcuPostCreateScenario_inch.tcl
USER with UG Standard seedpart	102	none	none
USER with ECU seedpart	121	EcuCreateBySeedPart.tcl	none

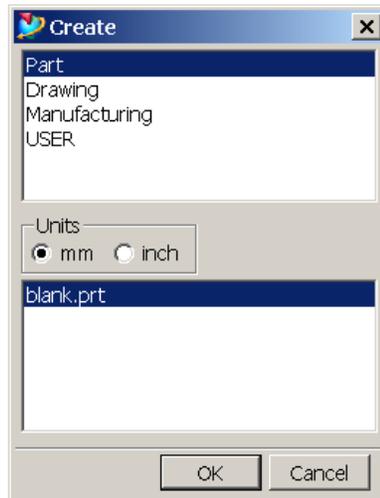
Embedding New Create Dialog and Seedparts

The new Create Dialog is activated by placing the following files into the ecu directories as follows:

File	Destination path	Purpose
ecu_cre.dlg	\$ECU_ROOT/application/\$ECU_UGV/application	UG dialog file
EcuCreateMenu.tcl	\$ECU_ROOT/tcl	ECU dialog logic
EcuCreateMenu.ini	\$ECU_ROOT/tcl	Seedpart Configuration file

The Seedpart configuration is done in EcuCreateMenu.ini. Please see the sample configuration and the resulting create dialog:

```
#####
# mapping file for seedparts
# XPLM Solution
# created ZIB 2006/05/31
#####
#
[ DefaultSettings ]
#
# default selection for UNIT switch, 0=mm, 1=inch
UNIT      = 0
# default selection for TYPE, zero based index of EcuCreateTypes
TYPE      = 0
#
[ EcuCreateTypes ]
#
# Object type = CAX subtype
#
Part       = NONE
Drawing   = DP
Manufacturing = MP
# Scenario = SP
USER      = USER
#
[ EcuSeedParts ]
#
# symbolic name      = ecu-internal seedpart
# {TYPE}_{UNIT}     = file.prt
Part_mm        = blank.prt
Part_inch     = blank_in.prt
Drawing_mm    = drw.prt
Drawing_inch  = drw_in.prt
Manufacturing_mm = cam.prt
Manufacturing_inch = cam_in.prt
Scenario_mm   = scenario.prt
Scenario_inch = scenario_in.prt
USER_mm      = blank.prt
USER_mm      = drw.prt
USER_mm      = cam.prt
USER_inch   = blank_in.prt
USER_inch   = drw_in.prt
USER_inch   = cam_in.prt
#
[ EOF ]
#
```



Section [EcuCreateTypes] describes all seedpart types the dialog will display in the type selection list and assigns a CAX subtype to each of them. The meaning of the CAX subtype is as follows:

CAX subtype	Description
NONE	No CAX subtype is set, this means the base part seedpart and trigger will be used.
DP	Drawing subtype, a drawing seedpart and trigger will be used
MP	Manufacturing subtype, a manufacturing seedpart and trigger will be used
SP	Scenario subtype, a scenario seedpart and trigger will be used
USER	The Part will be generated in local directory only, not inside PLM. If there is no seedpart defined in [EcuSeedParts] section, the UG standard create userexit is called. If there is a seedpart defined in [EcuSeedParts] section, the user can select it in dialog and the new part will be generated out of this seedpart.

Section [EcuSeedParts] defines all seedparts. The dialog filters the selectable seedparts by the currently selected type and unit inside the dialog. The symbolic seedpart name (left side) is build out of the type followed by a underscore and the unit. If there is more than one entry for the same symbolic name, the user can choose in dialog between more than one seedpart. The ecu-internal seedpart file name is given on the right side. There are 3 options, how the seedpart is found.

1. The environment variable `ecu_blank_part` defines the path to the part seedpart. If the seedparts in [EcuSeedParts] section doesn't contain the full path, the seedpart must be found in the directory beside the part seedpart.
2. The ecu-internal file name contains the full path.
3. If the seedpart file can't be found using option 1 and 2, the dialog uses the old environment variables for seedpart depending on the subtype definition as follows. The environment variables contain the full path to the seedpart.

Environment	Subtype environment (Standard)	Seedpart description
ecu_blank_part		Base Part metric
ecu_blank_part_in		Base Part english
ecu_blank_drw	cax_dp_ident=DP	Base Drawing metric
ecu_blank_drw_in	cax_dp_ident=DP	Base Drawing english
ecu_blank_cam	cax_mp_ident=MP	Base Manufacturing metric
ecu_blank_cam_in	cax_mp_ident=MP	Base Manufacturing english
ecu_blank_sce	cax_sp_ident=SP	Base Scenario metric
ecu_blank_sce_in	cax_sp_ident=SP	Base Scenario english

Default settings for unit or type of the seedpart can be configured in [DefaultSettings] section. The switches are as follows:

Switch	Description	Default
UNIT	Available values are: 0 - for metric/mm 1 - for english/inches	UNIT=0
TYPE	By default preselected value in type selection box. Available values are the zero based indexes of the types configures in [EcuCreateTypes] section. This value shouldn't be modified.	TYPE=0

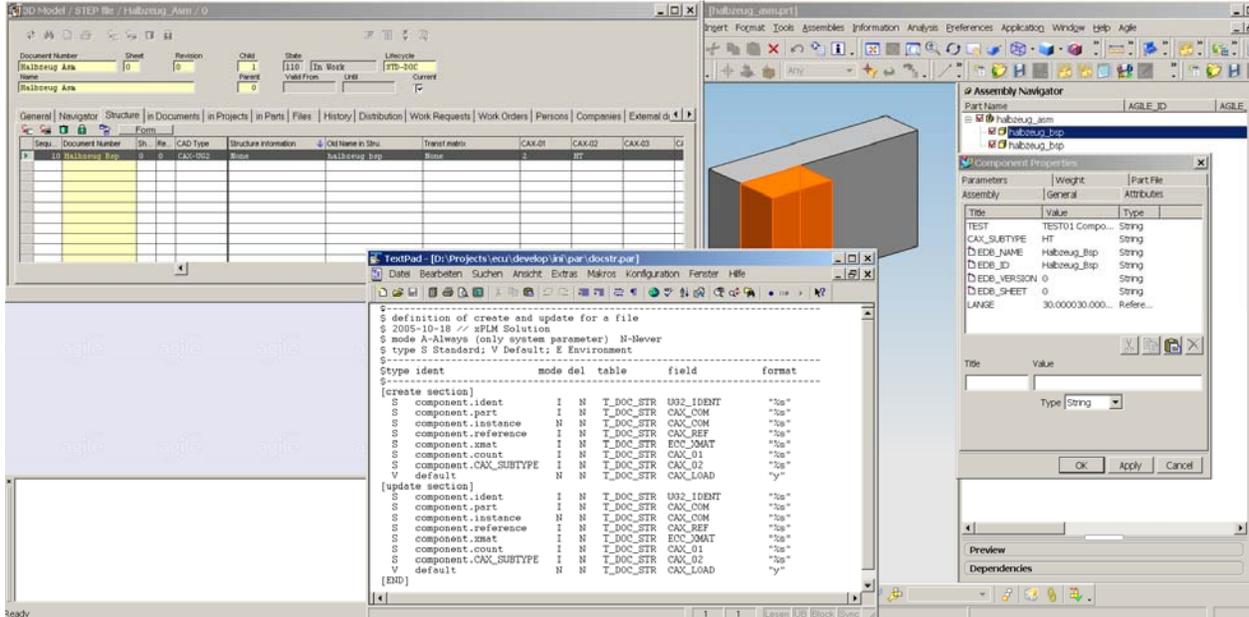
Transfer of Component Properties to PLM using docstr.par

The transfer of component properties to the document structure can be configured using the new docstr.par. If this file is missing, no component attributes are transferred to PLM structure. There are several new options for the par configuration files, to transfer internal attributes (like count, instance name, component name, reference set, position info) and

component attributes like shown in screenshot. The new options for docstr.par are as follows, all other options for *.par files may be used in docstr.par as well:

Option	Default field	Description
S component.ident	T_DOC_STR.UG2_IDENT (don't change)	CAX-Ident for this structure entry
S component.part	T_DOC_STR.CAX_COM (don't change)	Part name which is assigned to the component, required for renaming and STEP
S component.instance	T_DOC_STR.CAX_COM (optional instead of component.part)	Component name, required for old logic of assembly dependent auxiliary parts
S component.reference	T_DOC_STR.CAX_REF (don't change)	Component reference set
S component.xmat	T_DOC_STR.ECC_XMAT (don't change)	Encrypted transformation matrix, if STEP is enabled
S component.count	T_DOC_STR.CAX_01	Component count, if STEP is not enabled, transfer is optional
S component.{ATTR}	Project specific field in T_DOC_STR	Transfer of a project specific component property {ATTR} into a field. See example in next row
S component.CAX_SUBTYPE	T_DOC_STR.CAX_02	Example for assembly dependent auxiliary parts: Transfer of the component attribute CAX_SUBTYPE into the field T_DOC_STR.CAX_02 of the document structure.
S component.SUPPRESSED	T_DOC_STR.CAX_03	Mapping information if the child component is suppressed in NX

See sample assembly, configuration and resulting data in PLM structure:



Sample standard configuration in docstr.par for renaming (EcuRename=1) and/or STEP (EcuStepTransformation=1):

```

$-----
$ definition of create and update for a file
$ 2006-06-30 // xPLM Solution
$ mode A-Always (only system parameter) N-Never
$ type S Standard; V Default; E Environment
$-----
$type ident          mode del table      field      format
$-----
[create section]
S component.ident    I N T_DOC_STR UG2_IDENT    "%s"
S component.part     I N T_DOC_STR CAX_COM     "%s"
S component.reference I N T_DOC_STR CAX_REF     "%s"
S component.xmat     I N T_DOC_STR ECC_XMAT    "%s"
S component.count    I N T_DOC_STR CAX_01     "%s"
S component.CAX_SUBTYPE I N T_DOC_STR CAX_02     "%s"
V default           N N T_DOC_STR CAX_LOAD    "y"

[update section]
S component.ident    I N T_DOC_STR UG2_IDENT    "%s"
S component.part     I N T_DOC_STR CAX_COM     "%s"
S component.reference I N T_DOC_STR CAX_REF     "%s"
S component.xmat     I N T_DOC_STR ECC_XMAT    "%s"
S component.count    I N T_DOC_STR CAX_01     "%s"
S component.CAX_SUBTYPE I N T_DOC_STR CAX_02     "%s"
V default           N N T_DOC_STR CAX_LOAD    "y"

[END]
    
```

Sample alternate configuration in docstr.par for backward compatibility of assembly dependent auxiliary parts (transfer of component name instead of part name into T_DOC_STR.CAX_COM). This doesn't work with enabled renaming or STEP options!

```

$-----
$ definition of create and update for a file
$ 2006-06-30 // xPLM Solution
$ mode A-Always (only system parameter) N-Never
$ type S Standard; V Default; E Environment
$-----
$type ident          mode del table   field      format
$-----
[create section]
S component.ident    I  N  T_DOC_STR  UG2_IDENT    "%s"
S component.instance I  N  T_DOC_STR  CAX_COM      "%s"
S component.reference I  N  T_DOC_STR  CAX_REF      "%s"
S component.xmat     I  N  T_DOC_STR  ECC_XMAT     "%s"
S component.count    I  N  T_DOC_STR  CAX_01      "%s"
S component.CAX_SUBTYPE I  N  T_DOC_STR  CAX_02      "%s"
V default            N  N  T_DOC_STR  CAX_LOAD    "y"
[update section]
S component.ident    I  N  T_DOC_STR  UG2_IDENT    "%s"
S component.instance I  N  T_DOC_STR  CAX_COM      "%s"
S component.reference I  N  T_DOC_STR  CAX_REF      "%s"
S component.xmat     I  N  T_DOC_STR  ECC_XMAT     "%s"
S component.count    I  N  T_DOC_STR  CAX_01      "%s"
S component.CAX_SUBTYPE I  N  T_DOC_STR  CAX_02      "%s"
V default            N  N  T_DOC_STR  CAX_LOAD    "y"
[END]

```

Embedding configurable and project specific CAX Subtypes

Support of configurable CAX Subtypes requires modification of menufiles and initialisation files. The TCL Scripts EcuSetSubtype.tcl and EcuInit.tcl contain the logic and are required for execution of this menu item. Make sure you use the latest versions of both. The menufile agile.men / ecu.men needs an entry like this, that can replace all the other menu items for setting and removing of subtypes:

```

BUTTON PLM_SET_SUBTYPE
LABEL Set
ACTIONS plm_action

```

Configuration of the subtypes the user can choose when executing the "Set Subtype" menu item is done in tcl/Ecu.ini. The required sections have to be added to the tcl/Ecu.ini if not existing. The language specific configuration in tcl/Ecu.ini is done like this:

```

[ EcuSubTypes ]
#
Part      = NONE
Drawing   = DP
Manufacturing = MP
Auxiliary Part HT = HT
Auxiliary Part LT = LT
Standard Part = NT
#
[ EcuSubTypes_ger ]

```

```
#  
Teil      = NONE  
Zeichnung = DP  
CAM       = MP  
Hilfsteil HT = Hat  
Hilfsteil LT = LT  
Normteil  = NT
```

Large Assemblies – Add Component from PLM

Regeneration of displays after each add command took a long time in large assemblies. Now the component can be added hidden by not changing the display. Required files are EcuQuickLoad.tcl and Binaries as well as an entry in ufrcp.men, that refers the new hidden open component function no 120. The former function number 115 is still valid and working for compatibility. Example for the new setting:

```
PLM  
PLM  
help text  
help Text  
120
```

Large Assemblies – Suppress Loading of WAVE parents during save

Loading of WAVE parents may take a lot of time in large assemblies. This can be suppressed by setting the environment switch in tcl/Ecu.ini in [Initialize] section:

```
EcuWaveLoadParents = 0
```

Attention: If this option is set to 0, the adapter can't detect the link source of unloaded WAVE parents. Use ANT to make sure all the required WAVE parents are loaded, if you want to store the WAVE link information into PLM.