Agile
Version e6.0

ORACLE

Oracle Agile Engineering Data Management

Oracle Agile Engineering Data Management - MCAD Connector for Unigraphics NX Version 3.9.2.0 Installation and Administration Manual

Part No. E12428-01
April 2008

Copyright and Trademarks

The Programs (which include both the software and documentation) contain proprietary information; they are provided under a license agreement containing restrictions on use and disclosure and are also protected by copyright, patent, and other intellectual and industrial property laws. Reverse engineering, disassembly, or decompilation of the Programs, except to the extent required to obtain interoperability with other independently created software or as specified by law, is prohibited.

The information contained in this document is subject to change without notice. If you find any problems in the documentation, please report them to us in writing. This document is not warranted to be error-free. Except as may be expressly permitted in your license agreement for these Programs, no part of these Programs may be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose.

If the Programs are delivered to the United States Government or anyone licensing or using the Programs on behalf of the United States Government, the following notice is applicable:

U.S. GOVERNMENT RIGHTS Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the Programs, including documentation and technical data, shall be subject to the licensing restrictions set forth in the applicable Oracle license agreement, and, to the extent applicable, the additional rights set forth in FAR 52.227-19, Commercial Computer Software--Restricted Rights (June 1987). Oracle USA, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

The Programs are not intended for use in any nuclear, aviation, mass transit, medical, or other inherently dangerous applications. It shall be the licensee's responsibility to take all appropriate fail-safe, backup, redundancy and other measures to ensure the safe use of such applications if the Programs are used for such purposes, and we disclaim liability for any damages caused by such use of the Programs.

Oracle and Agile are registered trademarks of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

The Programs may provide links to Web sites and access to content, products, and services from third parties. Oracle is not responsible for the availability of, or any content provided on, third-party Web sites. You bear all risks associated with the use of such content. If you choose to purchase any products or services from a third party, the relationship is directly between you and the third party. Oracle is not responsible for: (a) the quality of third-party products or services; or (b) fulfilling any of the terms of the agreement with the third party, including delivery of products or services and warranty obligations related to purchased products or services. Oracle is not responsible for any loss or damage of any sort that you may incur from dealing with any third party.

CONTENTS

Preface

The Oracle documentation set includes Adobe® Acrobat™ PDF files. The Oracle Technology Network (OTN) Web site (http://www.oracle.com/technology/documentation/agile.html) contains the latest versions of the Oracle Agile EDM PDF files. You can view or download these manuals from the Web site, or you can ask your Agile administrator if there is an Oracle Documentation folder available on your network from which you can access the documentation (PDF) files.

Note To read the PDF files, you must use the free Adobe Acrobat Reader™ version 7.0 or later. This program can be downloaded from the Adobe Web site (http://www.adobe.com).

Note Before calling Agile Support about a problem with an Oracle Agile EDM manual, please have the full part number, which is located on the title page.

TTY Access to Oracle Support Services

Oracle provides dedicated Text Telephone (TTY) access to Oracle Support Services within the United States of America 24 hours a day, 7 days a week. For TTY support, call 800.446.2398. Outside the United States, call +1.407.458.2479.

Documentation Accessibility

Our goal is to make Oracle products, services, and supporting documentation accessible, with good usability, to the disabled community. To that end, our documentation includes features that make information available to users of assistive technology. This documentation is available in HTML format, and contains markup to facilitate access by the disabled community. Accessibility standards will continue to evolve over time, and Oracle is actively engaged with other market-leading technology vendors to address technical obstacles so that our documentation can be accessible to all of our customers. For more information, visit the Oracle Accessibility Program Web site at http://www.oracle.com/accessibility/ http://www.oracle.com/accessibility/.

Readme

Any last-minute information about Oracle Agile EDM can be found in the Release Notes file on the Oracle Technology Network (OTN) Web site
(http://www.oracle.com/technology/documentation/agile.html)

Agile Training Aids

Go to the Oracle University Web page
(http://www.oracle.com/education/chooser/selectcountry_new.html) for more information on Agile Training offerings.

Accessibility of Code Examples in Documentation

Screen readers may not always correctly read the code examples in this document. The conventions for writing code require that closing braces should appear on an otherwise empty line; however, some screen readers may not always read a line of text that consists solely of a bracket or brace.

Accessibility of Links to External Web Sites in Documentation

This documentation may contain links to Web sites of other companies or organizations that Oracle does not own or control. Oracle neither evaluates nor makes any representations regarding the accessibility of these Web sites.

Chapter 1

Introduction

Overview

The information in this document is based on a standard installation and the actual appearance of a specific installation may vary depending on the local environment and custom configuration realized for the individual requirements of the customer.

The integration functions are provided by additional and expanded menus in Agile e6 and Unigraphics which access the ECI interface for CAD and EDB system functions and the CAD and EDB database.

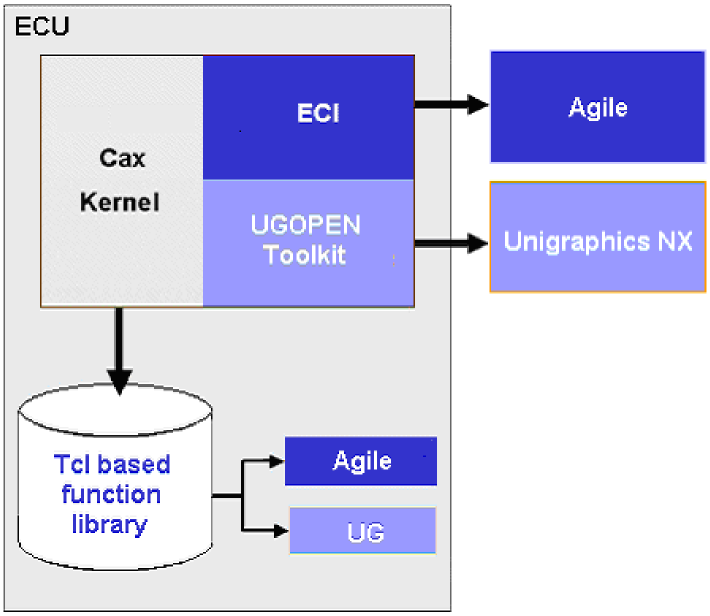
All management data are entered interactively in Agile e6 allowing a variety of document types.

Architecture

The Agile e6 Unigraphics Integration was created using the development tool UGOPEN. It represents an enhancement to the operation of Unigraphics. And is an add-on module to Agile e6.

An UGOPEN Execute License from UNIGRAPHICS SOLUTIONS is required for concurrently operating workstations

The architecture of the integration between Agile e6 and Unigraphics is illustrated in the following graphic:



Chapter 2

Installation

The installation of the Agile e6 Unigraphics Integration (ECU) consists of the following steps:

1. Prerequisites
2. Checking the Prerequisites
3. Unpacking the Archive File
4. Adapting the Startup Script
5. Customizing the Vault Options

Prerequisites

The Agile e6- Unigraphics integration is currently available for several hardware platforms and operating systems. Please check the Release Notes for an overview.

Checking the Prerequisites

- A database (e.g. ORACLE®) is installed and running.
- Agile e6 is installed and running inclusive FMS (File Management Server) or DFM (Distributed File Management) .
- A designated user environment exists in Agile e6.
- In Agile e6 a test user with a valid password exists and is authorized to start the Agile e6 session inclusive.
- A license for using the ECU integration is registered and available on the FELICS license server (you can obtain the ECU licenses from the Agile sales representative).
- Unigraphics is installed. The test user can start it from his home directory.

Unpacking the Archive File

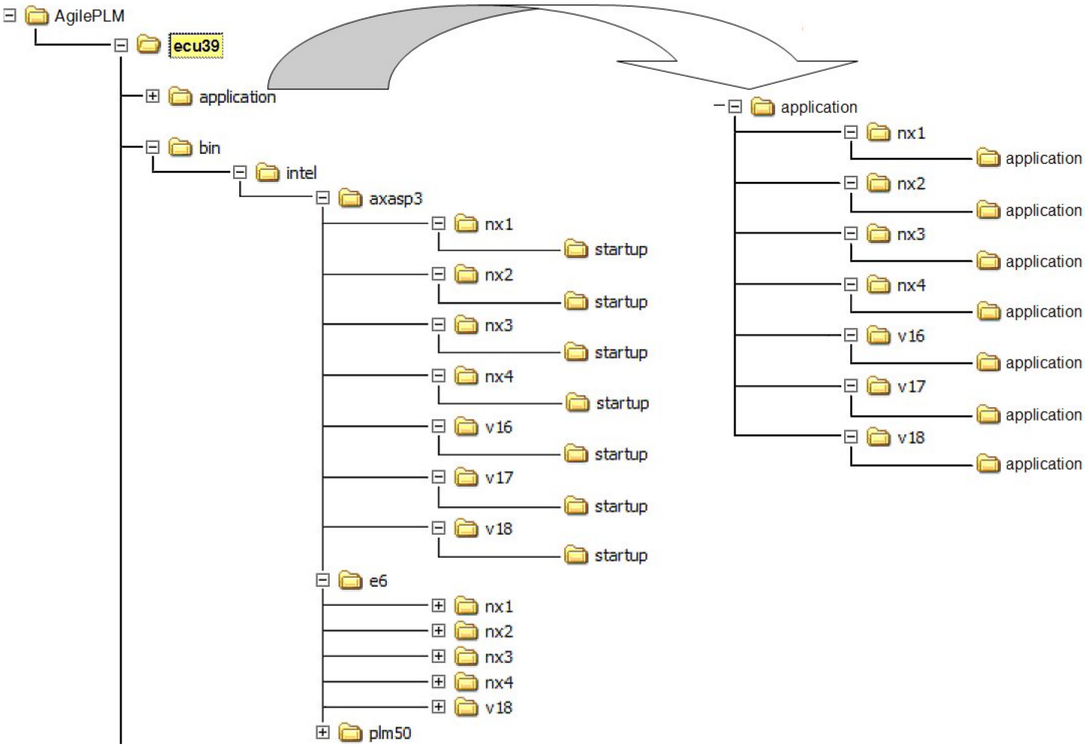
To install the ECU Integration, extract the delivered zip file to any path on your hard drive (e.g. C:\AgilePLM\ecu39).

Using .../ecu on a Unix platform, respectively ...\ecu on a MS Windows platform as the root directory (i.e. the value of the environment variable $ECU_ROOT) the following directory structure is created during installation:

Main Directory	Sub Directory	Sub Directory	Description
\application			Contains Unigraphics menu files (*.dlg) in binary format. For each supported Unigraphics version a separate sub-tree exists.
\bin			Contains program libraries.
	\hpux		Contains ECU libraries for HP-UX.
		\plmX*	Contains libraries used by the EDB System Agile e6 (e.g. \e6).
		\axaspX*	Contains libraries used by the EDB System axalant® (e.g. \axasp3).
	\hpux64		Contains ECU libraries for 64bit version of HP-UX.
		\plmX*	Contains libraries used by the EDB System Agile e6 (e.g. \e6).
	\intel		Contains ECU libraries for MS WINDOWS NT.
		\plmX*	Contains libraries used by the EDB System Agile e6 (e.g. \e6).
		\axaspX*	Contains libraries used by the EDB System axalant® (e.g. \axasp3).
	\solaris		Contains ECU libraries for SUN-OS.
		\plmX *	Contains libraries used by the EDB System Agile e6 (e.g. \e6).
		\axaspX*	Contains libraries used by the EDB System axalant® (e.g. \axasp3).
	\solaris64		Contains ECU libraries for 64bit version of SUN-OS.
		\plmX *	Contains libraries used by the EDB System Agile e6 (e.g. \e6).
\com			Contains scrypts of the integration.
\frames			Contains examples for drawing frames.
\ini			Contains initialization files and informational text.
	\fra		
	\par		
	\e6	\eng \ger	Contains additional menu text if the EDB System Agile e6 is used (containing an English and a German sub tree).

Main Directory	Sub Directory	Sub Directory	Description
	\axaspX	\eng \ger	Contains additional menu text if the EDB System axalant is used (containing an English and a German sub tree).
	\plmX	\eng \ger	Contains additional menu texts if the EDB System Agile e6 is used (containing an English and a German sub tree).
\tcl			Contains TCL procedures.
\plt			Contains procedures to enable plotting.
	\SDI		Contains special files for new SDIplot method.
		The character 'X' stands for a specific version or service pack (e.g. plm50; axasp3 etc.). The directories contain the version-specific files, mostly binaries or menu files.	

The Agile e-series directories also contain subdirectories for each supported Unigraphics version as displayed in the following figure for an Intel-MS Windows path.



Adapting the Startup Script

To set the customer specific variables, the respective startup file ecu_start.bat (Windows) in the directory %ECU_ROOT%\com needs to be adapted. If the specified paths are incorrectly set than the needed additional menus are not visible.

Ecu_start.bat

Variable	Description	Valid content
ECU_UGV	The used Unigraphics version	"v18", "nx1", "nx2", "nx3", "nx4" ["v16", "v17", this versions are unsupported]
EDB_DBS	The used Agile PLM version	"e6", "e5", "plm", "axa", "agile9" ["cadim", this product is unsupported]
ECU_LANG	Language of the ECU connector's menu tree	eng, ger
UGII_BASE_DIR	Unigraphics directory	D:\CAD\UGNX30
UGII_ROOT_DIR	Directory of the binary file ugraf.exe to run Unigraphics	D:\CAD\UGNX30\UGII
ug_start	Full name of the binary file to start Unigraphics	D:\CAD\UGNX30\UGII\ugraf.exe
EDB_CLI_DIR	Directory which contains the command procedure plm.cmd	C:\Agile_e6\axalant\cmd
EDB_PRF	Client profile	a string (-) (i.e. in most cases there is no profile established)
EDB_APP	Name of the DataView environment	A string (CAX)
EDB_SRV	The server name where the DataView server application is running	A string (localhost)
cax_usr_home	Device name where the local user directory ecu\% USERNAME% is located	D:

In Windows the start script ecu_start.bat calls the command procedure ug2.bat (Unix: ug2.ksh). Additional, internal command procedures are set that use variables deriving from the used Agile version defined in variable EDB_DBS.

Variable	Description	Valid content
EDB_COM	The internally used communication type	TCP
UGII_CUSTOM_DIRECTORY_FILE	The customer-specific file in Unigraphics	%ECU_ROOT%\ini\custom_dirs.dat If another customer-specific file is named here then that file must contain at least the following 3 lines: $ECU_ROOT/application/$ECU_UGV $ECU_ROOT/ini/${EDB_VER}${ECU_ADMIN}/${ECU_LANG} $ECU_ROOT/bin/$ECU_MACH/$EDB_VER/$ECU_UGV

The used CAX types are set according to the delivered PLM standard dump. The string "3D_MODEL" is always used for a cax_type. This is defined in the first lines of the file ECU_ROOT%\ini\ecu.ini (in Windows) resp. $ECU_ROOT/ini/ecu.ini (in Unix):

```
cax_type PRT
     3D_MODEL  TYP2
end

cax_type DRW
     3D_MODEL  TYP1
end

cax_type XPK
     3D_MODEL  TYP3
end
```

Note Because Unigraphics drawing files do not have a separate file extension, some customers want to organize their PLM system other like proposed in the standard dump. In this case the file ecu.ini has to be customized by a project manager.

Customizing Vault Options

To run the out-of-the-box configuration of the ECU connector it is not necessary to adapt the data model of the PDM system.

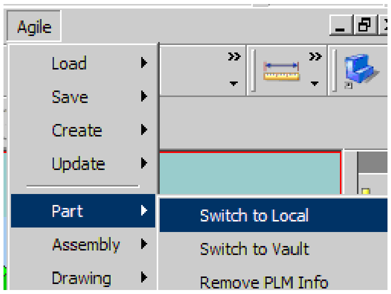
Ensure that the message entries ECU-ASG-PRT and ECU-STO-PRT exist in the message list (table T_MESSAGE).

Note If the messages do not exist, they have to be created manually (PLM-user = EDBCUSTO).

Working with the Agile e6 FMS (File Management Server) ensures centralized and secure storage of file objects.

It is always possible to specify whether the actual object should be saved externally or in the file server's vault.

- ❑ Local > FilServ switches from "local" work to file server
- ❑ FilServ > Local switches from file server to "local" work



Note How to set up a vault can be found in the Agile e6 documentation.

FMS Entries

Before the configured file server can be used the vault name must be known in the integration. Therefore, the file %ECU_ROOT%\ini\par\docfile.par must be edited. The string "CADFILES" in the first line of section [create section] has to be replaced by the vault name to be used.

The following uses the vault name "UGFILES":

default: V Default_Area I N T_FILE_DAT STORAGE_AREA "CADFILES"

adapted: V Default_Area I N T_FILE_DAT STORAGE_AREA " UGFILES "

Note If Distibuted File Management (DFM) is used than fully delete this line, but avoid blank lines.
Alternatively it is possible to change in this line the third parameter from "I" to "N".
The file docfile.par_dfm is an example script for DFM.

Check the Files in the File Server:

1. The actual object is written into the user temp directory %cax_temp%.
2. The file server reads the file and copies it with an encoded name into the vault.
3. If the file has been successfully checked-in it is deleted in the temp directory.

Note To suppress this step, set the value of cax_no_del_fs to "1" in the file ECU_INI.BAT.

Chapter 3

Call Procedure when Starting ECU

Call Procedure for MS Windows and UNIX

The called procedure is almost analog in MS Windows (*.bat files) and UNIX (*.ksh files) after starting the ecu_start file.

The procedure of called files looks as following:

	MS Windows	UNIX	Implication
1.	ecu_start.bat	ecu_start.ksh	All version specific switches will be set here. This will be arranged automatically during the installation procedure. Subsequent, the command procedure ug2.bat is called.
2.	❑	ug2_env.ksh	Settings which are specific to the particular Unix platforms.
3.	ug2.bat	ug2.ksh	The version of the ECU integration and of the EDB is defined. Further switches will be set in the procedure ecu_ini.bat, which is called from inside this procedure. The path is then extended with ECU specific directories, and the transport protocol is defined. Subsequent, the startup command for Agile e6 (defined in edb_start.bat) and Unigraphics will be activated.
4.	ecu_ini.bat	ecu_ini.ksh	Additional ECU specific switches will be set here. This is described in the following examples.

The following examples specify the delivered content of the 3 files running in MS Windows in detail.

ecu_start.bat

This command procedure starts a modified Unigraphics session. The UG main menu is extended by the Agile submenu. Additionally, some UG-native buttons are extended too (e.g. the open file command).

Command Line	Description
@echo off	
set CUR_DIR=%~dp0%	Current command procedure directory of the ECU integration.
set ECU_ROOT=%CUR_DIR:~0,-5%	Current directory of the ECU integration.

Command Line	Description
rem Version Settings:	
set ECU_VER=39	Version of the ECU integration (works with the correct DLL).
set ECU_UGV=nx3	Unigraphics version (e.g. v18, nx1, nx2, nx3, nx4).
set EDB_DBS=e6	Agile version (e.g. e6, e5, plm, axa, agile9).
set ECU_LANG=eng	Language of the menus (e.g. eng, ger).
rem UGII Settings:	
set UGII_ROOT_DIR =D:\EDS\UGNX\UGII	Root directory of Unigraphics.
set ug_start D:\EDS\UGNX\UGII\ugraf.exe	Name of the procedure to start up Unigraphics.
rem EDB Settings:	
set EDB_CLI_DIR =D:\AgilePLM\ext\bin\intel-ms-nt4.0\	Directory containing the binaries used by the Agile e6 client.
set EDB_APP=cax	Name of the EDB application to be used (- means "nothing set").
set EDB_SRV=server1	Name of the server the EDB application is running on (“-“ = "nothing set").
rem User Home Settings:	
set cax_usr_home=D:	Device name of the local user directory ecu\%USERNAME% (used for saving the temporary data files of the ECU integration).
call %ECU_ROOT%\com\ug2.bat %*	
exit	

ug2.bat

This command procedure initializes the internal settings for the ECU integration, depending on the used PLM system. Only the settings for Agile e6 are described here.

Command Line	Description
@echo off	

Command Line	Description
rem Switch the systems	Parameter settings depending on the used EDB software. It is essential to use the correct binaries and ECI transport protocols.
if "%ECU_VER%" == "" set ECU_VER=39	Default value of the ECU version.
@echo off	
if "%EDB_DBS%" == "e6" goto e6	
:e6	Internal settings for Agile e6.
set EDB_VER=e6	Default value of the EDB version.
set EDB_COM=TCP	Default value of the transport protocol.
call %ECU_ROOT%\ini\ecu_ini.bat	This command procedure is described in the next section.
set UGII_BASE_DIR=%UGII_ROOT_DIR%\..	Default directory of the Unigraphics base directory.
set PATH=%EDB_CLI_DIR%; %EDB_CLI_DIR%\..\..\axalant\bin\intel-ms-nt5.0; %ECU_ROOT%\bin\intel\%EDB_VER%; %UGII_ROOT_DIR%; %PATH%	Enlarging the Windows search path
goto ug_startup	
:ug_startup	UGII environment.
chdir /D %cax_temp%	%cax_temp% is set in file %ECU_ROOT%\ini\ecu_ini.bat (see next section).
set UGII_CUSTOM_DIRECTORY_FILE=%ECU_ROOT%\ini\custom_dirs.dat	This parameter is initialized with the name of a file containing special UGII environment settings for customer specific embeddings. The syntax of entries in that file always has to be UNIX-like.
:ugraf_startup	
if not "%EDB_DBS%" == "agile" call %edb_start%	% edb_start % is set in file %ECU_ROOT%\ini\ecu_ini.bat (see next section).

Command Line	Description
start %ug_start%	% ug_start % is set in file %ECU_ROOT%\com\ecu_start.bat (see previous section).
goto end	
:end	

ecu_ini.bat

ECU specific switches can be set here.

Note A blue colored "rem" indicates that the setting must NOT be activated in the application, because some other customizations are necessary inside of the repository. For more detailed information please refer to your Project Manager.

Command Line	Description
if "%EcuUseCollaboration%" == "1" set ECU_COL=_c	For internal use only.
set ecu_bin=%ECU_ROOT%\bin\intel\%EDB_VER%	Sets the directory name for the used ECU connector binaries.
set ecu_lib=%ecu_bin%\ecu_%ECU_VER%_%ECU_UGV%.dll	Sets the name of the used ECU dynamic link library.
set ecu_exe= %ecu_bin%\ecu_%ECU_VER%_%ECU_UGV%%ECU_COL%.exe	Sets the name of the used ECU executable file.
set ecu_ini=%ECU_ROOT%\ini	Sets the name for the ECU subdirectory …\ini.
set ECU_ADMIN=	Enables the creation of user specific menu adaptations.
set ecu_txt=%ecu_ini%\%EDB_VER%%ECU_ADMIN%	Path of the user specific menu adaptation.
rem edb variables	
rem ecu variables	
if "%ECU_LANG%" == "ger" set ecu_oth_lan=1	For internal use only. Sets the German language for the logfile content.

Command Line	Description
set pro_par_ini=%ecu_ini%\par\	Sets the name for the ECU subdirectory …\ini\par.
set cax_trn=%ecu_ini%\ecu.trn	For internal use only. Set the behaviour when translating non ASCII characters.
set cax_ini=%ecu_ini%\ecu.ini	Sets the name of the used ECU main initialisation file.
set edb_cax_dis=%cax_usr_home%	Local directory for temporary CAD user data (can be adapted).
set edb_cax_dir=\ecu\%USERNAME%\tmp	User specific directory for temporary CAD user data (can be adapted).
set cax_temp=%edb_cax_dis%%edb_cax_dir%\	User specific directory for temporary CAD user data.
set edb_cax_nt_dis=%ECU_ROOT%	For internal use only. Sets device name of the norm part directory.
set edb_cax_nt_dir=\fast\	For internal use only. SetS the norm part directory.
set edb_cax_nod=%COMPUTERNAME%	For internal use only.
set ecu_frm=%ECU_ROOT%\frames	Sets the name for the ECU subdirectory …\frames.
set cax_allow_dmf=1	
…	Commented lines should only be activated when special customizing was made before.
rem eci variables rem	Do not change the following lines!
rem define ECI topic	Definition of the ECI topic.
if "%EDB_COM%" == "DDE" (	The next 6 lines are only set when using DDE.
set cax_node_name=%COMPUTERNAME%	For internal use only.
set cax_trnspt=57	For internal use only.
set cax_num=%COMPUTERNAME%	For internal use only.
set cax_sock=ECU_ON_	For internal use only.
set ECI_DDE_SYNCHRON=0	For internal use only.

Command Line	Description
set ECI_DDE_TIMEOUT=2000)	For internal use only.
else (	The next 4 lines are set when not using DDE.
set cax_node_name=localhost	For internal use only.
set cax_trnspt=1	For internal use only.
set cax_num=9	For internal use only.
set cax_sock=999)	For internal use only.
set cax_soc=%cax_sock%%cax_num%	For internal use only.
rem CaxDelay wait for eci in ms	
set CaxDelay=30	For internal use only. Only when using DDE
set dd_sleep=20	For internal use only.
set autostart=1	For internal use only.
rem example to switch EDB user	
set USRNAM=%USERNAME%	Username given to e6 client for login.
if "%USERNAME%"=="administrator" set USRNAM=edbcusto	Provide certain EDB users with administration rights.
if "%USERNAME%"=="Administrator" set USRNAM=edbcusto	
rem Plotting mode	
set cax_plt=%cax_temp%plt\	Sets the name for the ECU plot directory …\plt
set cax_plt=%ECU_ROOT%\plt\	Overwrites the name for the ECU plot directory …\plt
rem Menu control	
set ecu_men_ini=%ecu_txt%\men_%ECU_LANG%.ini	Only used for backward compatibility. Sets the name of the used ECU old menu initialisation file.
set cax_msg_txt=%ecu_txt%\ecumsg.txt	Sets the name of the used ECU message text file.
set cax_men_txt=%ecu_txt%\	Sets the name for the ECU menu text directory.

Command Line	Description
set USER_RETRIEVE=%ecu_lib%	Enables the use of a special open function in the Agile ecu-binary.
set USER_CREATE=%ecu_lib%	Enables the use of a special create function in the Agile ecu-binary.
set USER_FILE=%ecu_lib%	Enables the use of a special save function in the Agile ecu-binary.
set USER_SAVEAS=%ecu_lib%	Enables the use of a special save as function in Agile ecu-binary.
set USER_STARTUP=%ecu_lib%	Enables the use of a special function in the Agile ecu-binary for initializing and batchplotting.
set USER_RCOMP=%ecu_lib%	Enables the use of a special retrieve component function in the Agile ecu-binary.
set USER_CCOMP=%ecu_lib%	Enables the use of a special create component function in the Agile ecu-binary. This does not support the usage of seedparts. If seedparts are needed, this switch should be commented out.
set USER_SCOMP1=%ecu_lib%	Enables the use of a special substitute component function in the Agile ecu-binary.
set USER_SCOMP2=%ecu_lib%	Enables the use of a special substitute component function in the Agile ecu-binary.
rem work directory : where locally filed objects are kept	
set cax_work=%cax_usr_home%\ecu\%USERNAME%\work	Work directory to store objects locally.
set pro_cax_dis=%cax_usr_home%	
set pro_cax_dir=\ecu\%USERNAME%\work	Sets the subdirectory \work of the local user directory ecu\% USERNAME%.
set cax_fra_ini=%ecu_ini%\ecu_fra.ini	Frame initialization file.
set cax_bckup_dir=%cax_usr_home%\ecu\%USERNAME%\bck\	If the save function fails, the file will be stored here.
set cax_ign_cmp_nam=1	For internal use only.

Command Line	Description
set cax_ignore_step_str=1	For internal use only.
set ECP_ROOT=%ECU_ROOT%	For internal use only.
rem if set to 1 user files will be removed after checkin	
set cax_remove_user_file=0	If this function is set to 1, the user files will be removed after checkin.
rem OS Commands	
set cax_rm_cmd=%ECU_ROOT%\com\rm.com	For internal use only.
set cax_cp_cmd=copy	For internal use only.
set cax_rm_cmd=del	For internal use only.
set cax_mv_cmd=ren	For internal use only.
set cax_fil_chr=_	For internal use only.
set ECU_SEL_DRW_DEF=1	For internal use only.
set ECU_SEL_PRT_DEF=1	For internal use only.
set ecu_opn_top_quick=1	For internal use only.
rem checking working directories	For internal use only.
if not exist %cax_usr_home%\ecu\%USERNAME% mkdir %cax_usr_home%\ecu\%USERNAME%	Creates the required directories for ECU.
if not exist %cax_usr_home%\ecu\%USERNAME%\work mkdir %cax_usr_home%\ecu\%USERNAME%\work	Creates the required directories for ECU.
if not exist %cax_usr_home%\ecu\%USERNAME%\tmp mkdir %cax_usr_home%\ecu\%USERNAME%\tmp	Creates the required directories for ECU.
if not exist %cax_usr_home%\ecu\%USERNAME%\bck mkdir %cax_usr_home%\ecu\%USERNAME%\bck	Creates the required directories for ECU.

Command Line	Description
rem ---	
set ecu_diff_wrk_spc=True	For internal use only.
set edb_cax_dis_fs=%edb_cax_dis%	For internal use only.
set edb_cax_dir_fs=%edb_cax_dir%	For internal use only.
rem ---	
set ecu_blank_part=%ecu_ini%\blank.prt	Seedpart for Models (DIN).
set ecu_blank_part_in=%ecu_ini%\blank_in.prt	Seedpart for Models (INCH).
set ecu_blank_drw=%ecu_ini%\drw.prt	Seedpart for Drawings (DIN).
set ecu_blank_drw_in=%ecu_ini%\drw_in.prt	Seedpart for Drawings (INCH).
rem	
set cax_dp_ident=DP	Ident in database for Drawing parts.
set cax_np_ident=NT	Ident in Database for Standard parts.
set cax_hp_ident=HT	Ident in Database for Dummy parts.
rem LGV	
set cax_lgv_men=%ecu_ini%\lgv.ini	Sets the name of the used ECU old initial LogiView menu settings.
set cax_lgv_dat=%cax_usr_home%\ECU\%USERNAME%\lgv_dat	Used for backward compatibility
rem version 3.3.0	Available since ECU version 3.3.0.
rem necessary for part families	Controls part families.
set cax_no_del_fs=1	
set cax_remove_user_file=0	
rem version 3.3.1	Available since ECU version 3.3.1
rem necessary for TCL Support	
set CaxTclScripts=%ECU_ROOT%\tcl\	Path to ECU TCL-Procedures.
set CaxConfigDir=%ECU_ROOT%\tcl\	Path to ECU TCL-Mapping-files.

Command Line	Description
set TCL_LIBRARY=%ECU_ROOT%\tcllib	Path to ECU TCL-Procedures-Library.
rem version 3.3.2.3	Available since ECU version 3.3.2.3
rem no selection menu at saving	
rem set EcuNoPartMenu=True	For internal use only.
rem Check if Modified	
rem set EcuModifiedPart=True	For internal use only.
rem version 3.3.3.0	Available since ECU version 3.3.3.0
rem call for DataView-Start	Defines the command for EDB-Start.
set edb_start=%ECU_ROOT%\com\edb_start.bat	Defines the Startscript for PLM client.
rem version 3.4.0.0	Available since ECU version 3.4.0.0
rem set BCT_IDENT=BCT-BEZ1	Name of part attribute to recognize a BCT-standard part automatically. Enabled for BCT support only by removing the REM.
rem version 3.4.0.5	Available since ECU version 3.4.0.5
set CAX_SOCK=%cax_sock%	Ensures compatibility between connector implementations under NT4/NT5/UNIX
set CAX_NUM=%cax_num%	Ensures compatibility between connector implementations under NT4/NT5/UNIX
set CAX_TRNSPT=%cax_trnspt%	Ensures compatibility between connector implementations under NT4/NT5/UNIX
set NODE=%edb_cax_nod%	Ensure compatibility between connector implementations under NT4/NT5/UNIX
rem version 3.4.0.6	Available since ECU version 3.4.0.6
set CAX_TEMP=%cax_temp%	Ensures compatibility between connector implementations under NT4/NT5/UNIX
set CAX_WORK=%cax_work%	Ensures compatibility between connector implementations under NT4/NT5/UNIX
rem set EcuCreateInInteractiveMode=1	Creates new PLM objects always interactively if enabled, otherwise by the chosen save option.
rem version 3.4.0.7	Available since ECU version 3.4.0.7

Command Line	Description
rem BOM stop at help part	
rem version 3.4.0.11	Available since ECU version 3.4.0.11
rem changes the update mode for the template member structure	
set EcuUpdateFamily=1	Changes the update mode for the template member structure.
rem suppress the message boxes in batch plot mode	
rem set ecu_no_msg_box=1	Suppresses the message boxes while batch plot mode is running if REM is removed.
rem version 3.4.1	Available since ECU version 3.4.1
rem update the template table of part families with the member part attributes	
rem set EcuUpdateFamilyTemplate=1	If activated the template table of part families will be updated with the member part attributes if REM is removed.
rem search in EDB if the member already exists	
rem set EcuSearchFamilyMember=1	Searches in database whether a member already exists if REM is removed.
rem set the column to be displayed in member list	
rem set EcuFamilyMemberColumn=SF_BENE1	Change the displayed column in the member list if REM is removed.
rem partname prefix in template table for parts where the member partname has to be set up by EDB	
rem set EcuFamilyMemberCreateByEdb=edb	Part name prefix in template table for parts where part name has to be set up by EDB. Internal use only.
rem create/update only structure elements of selected family members, other structures will not be deleted	
rem set EcuOnlySelectedMembers=1	Creates/updates only structure elements of selected family members. All the other structures will not be deleted if REM is removed.

Command Line	Description
rem we get the first found article in frames if there is more than 1.	Returns the first found article in frames if there is more than 1.
rem set ecu_exact_one_article=1	We should break if there are more than 1 articles assigned set the following if REM is removed.
rem cax_art_lis=EDB-ART-SLI	Chnges the default item list if REM is removed
set tcl_plot=1	Use EcuBatchQueue.tcl for processing batch plots when starting ecu-lib with “Execute->Userfunction”.
rem version 3.5.0.1	Available since ECU version 3.5.0.1
rem count and fill occurrences of childs in T_DOC_STR.QUANTITY	
rem set EcuDocumentQuantity=1	Counts and fills occurrences of child records in T_DOC_STR.QUANTITY if REM is removed.
rem write T_DOC_STR.CAX_LOAD with y	
rem set CaxLoadFlag=1	Sets T_DOC_STR.CAX_LOAD to “y” if REM is removed.
rem version 3.5.0.1	
rem multiple article in frames	
set ecu_multiple_article=***	If more than one article is attached to one drawing, the text will be written into the frame.
rem frames	
rem set ecu_upd_fra=1	If activated the drawing frames will be updated after load
rem set cax_dwg_jk=1	For internal use only.
rem eof	

Chapter 4

Menu Structure

Note If it is necessary to adapt the standard menu text, it is recommended to save a copy of the original menu files before starting the modifications.

The required files are located in the directory %ECU_ROOT%\ini\%EDB_DBS%, and the language specific subdirectories.

Filename	Description
ufcre.men	Creates menu.
ufget.men ufget.men.Norm	Opens menu, obsolete if used Styler Dialog.
ufput.men	Saves menu, obsolete if used Styler Dialog.
ufrcp.men	Inserts component menu.
ufsvas.men	Save As submenu, obsolete if used Styler Dialog.

ecumsg.txt	Message texts in English and German:
	ECI = messages of the ECI interface
	ECU = messages of the ECU integration

ufccp.men	Creates component menu.

The English menu text and integration specific toolbars are located in the subdirectory eng\startup. The German menu text and integration specific toolbars are located in the subdirectory ger\startup. All bitmaps called by the toolbar description files *.tbr are located in the directory %ECU_ROOT%\application.

When starting Unigraphics NX3, only menus in menuscript format are supported.

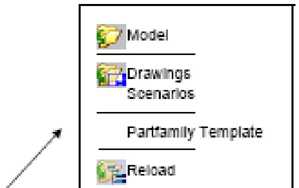
The menu structure of the integration has been unified. An appropriate menu structure exists for ECU, called the harmonized menu. The harmonized menu file agile.men is delivered with the integration and can be found in the directory %ECU_ROOT%\ini\%EDB_DBS%\eng\startup.

The previous menu structure is also delivered with the integration, but renamed to ecu.men_. Each startup subdirectory contains the respective files.

ECU 3.9.0

Agile Menu

Harmonized menu structure



ıgement

Chapter 5

Parameter Transfer

This chapter describes the data transfer from Unigraphics to Agile e6 and vice versa.

Transfer from Unigraphics to Agile e6

The Unigraphics parameters are transferred from the currently loaded object automatically to Agile e6 each time the object is saved. The parameters of the currently loaded object are always passed to the corresponding fields of the document. If no data record exists for the current object in Agile e6, these parameters are used as defaults for subsequent data records. This allows to automatically save unknown objects in Agile e6 in the background ("in batch mode").

The assignment of a Unigraphics parameter to the corresponding Agile e6 data record field is defined in so-called “parameter files” *.par. These files can be found in the directory that was defined in the file ecu_ini.bat. It is the value of the variable "pro_par_ini". The default is %ECU_ROOT%\ini\par\ (in MS Windows) and $ECU_ROOT/ini/par/ (in Unix).

The following parameter files can be found:

Parameter File	Description
3d_model.par	Document type 3D model.
3d_modeldp.par	Document subtype drawing (master/model).
item.par	Item type.
docfile.par	Definition of field content of the document file assignment when no DFM is used for check-in
docfile.par_dfm	Example for definition of field content of the document file assignment when DFM is used for check-in. The decisive difference is the mode of "Default_area". It has to be "N" in order to allow DFM to specify the storage area.
docstr.par	Component properties related to the document structure

Additional parameter files, that are also located in this directory, are used to define subtypes, and for saving documents with additional information. The file name of these parameter files is composed from the content of the PLM-fields T_DOC_DAT.DOC_TYPE and T_DOC_DAT.CAX_SUBTYPE, e.g. 3d_modelep. This enables to customize the parameter transfer for each particular object type.

Note Parameter transfer can be used as needed or suppressed by simply renaming the parameter files.

Parameter files are not essential for a correct working of the software interface.

It is possible to specify the parameters more than once if identical data is used.

Note Avoid blank lines in the parameter file. This may cause major instabilities of the integration.

Note Ensure that all listed fields are defined in the corresponding Agile e6 masks. The fields must be of sufficient length. It is also important to adhere to the corresponding format definition in the "format" column.

Description of Parameter Entries

The parameter files have a unique and logical structure:

Header	Drawing.par: standard ecu default values for documents Modus A-Always / I-If exist (User parameter only) / N-Never S Standard / A Attribut / P &Parameter / D Dimension / V fixed value Typ Ident Modus Delete Tabelle Feld Format						
	Typ	ident	mode	del	table	field	format
	S	drw_size	A	N	T_DOC_DAT	CAX_FRAME_ID	"%s"
	S	sheet_cnt	A	N	T_DOC_DAT	SHEET_NO	"%d"
	S	obj_typ	A	N	T_DOC_DAT	CAX_TYPE	"%s"
	S	obj_nam	N	N	T_DOC_DAT	DOCUMENT_ID	"%B"
	S	obj_nam	A	N	T_DOC_DAT	DOC_NAME	"%s"
	S	obj_nam	A	N	T_DOC_DAT	CAX_FIL_OLD_NAME	"%s"
	S	drw_frm_typ	A	N	T_DOC_DAT	CAX_SUBTYPE	"EP"
	S	family_index	A	N	T_DOC_DAT	CAX_VAR	"%s"
	S	version	A	N	T_DOC_DAT	CAX_CRE_SYSTEM	"%s"
	V	loc_flg	A	N	T_DOC_DAT	CAX_LOCAL	"n"
	S	part.NAME	A	N	T_DOC_DAT	DOC_NAME	"%s"
	C	DOCUMENT_ID	A	N	T_DOC_DAT	DOCUMENT_ID	"%s"

	S	drw_size	A	N	T_DOC_DAT	CAX_FRAME_ID	"%s"
	S	family_index	A	N	T_DOC_DAT	CAX_VAR	"%s"
	S	version	A	N	T_DOC_DAT	CAX_CRE_SYSTEM	"%s"
	S	obj_nam	A	N	T_DOC_DAT	CAX_FIL_OLD_NAME	"%s"
	C	DOCUMENT_ID	N	N	T_DOC_DAT	DOCUMENT_ID	"%s"
	C	DOCUMENT_ID	N	N	T_DOC_DAT	DOCUMENT_ID	"%s"

A header consists of comment lines and contains several information such as file description and change date. It is followed by lines with short parameter descriptions.

Table entries describe the actual parameters. They are divided into a [create section] and an [update section].

The "create section" is processed whenever a new record is created, while the "update section" is only relevant for updates of records. The parameter entries defined in one of these sections are transferred when the section is processed.

Note The upper limit is 32 entries per section!

A detailed description of the entries is provided below.

Column	Value	Description	
Type:	❑	Defines the parameter type	
❑	A	Attribute	Format: %s
❑	C	Agile e6 data	Use of subordinate document data for items and drawings. Agile e6 data can be inherited from (PRT/ASM) drawing models and items to drawings (DRW). Syntax: - field of the document master table or with preceding period: <.Field> - field of the document type table Format: %s
❑	E	Environment	Content of the specified environment variable.
❑	V	Fixed value	Fixed value Format: “value”

Column	Value	Description	
❑	S	Standard parameter	Predefined parameter Format: various
ident:	❑	Identification of the parameter	
❑	%s	version	Current source system version
❑	%d	sheet_no	Sheet number
❑	%s	drw_size	Drawing format of the current object
❑	y/n	loc_flg	Default local flag (system parameter)
❑	%f	mass_g(*)	Mass of the current object in gram
❑	%f	mass_kg(*)	Mass of the current object in kilogram
❑	%s	obj_nam	UNIGRAPHICS name of the current object
❑	%s	obj_typ	UNIGRAPHICS type of the current object (system parameter)
❑	%s	cre_sys	UNIGRAPHICS version
❑	%f	surface_m (*)	Surface of the current object in square meter
❑	%f	surface_mm (*	Surface of the current object in square millimetre
❑	%s	unit_l	Measurement units for length of the current object
❑	%s	unit_m	Measurement units for mass of the current object
❑	%f	volume_m(*)	Volume of the current object in cubic meter
❑	%f	volume_mm(*)	Volume of the current object in cubic millimetre
❑	%f	drw_frm_typ	Subtype
❑	%s	family_index	Identification for family table parts
❑	%d	sheet_cnt	Total sheet number
❑	%s	fil_nam	Name of the file which is actually processed
❑	%s	part.<Attr>	UG Partattribute <Attr> in actual partfile
mode:	❑	Defines the execution mode	

Column	Value	Description	
□	A	Always	Always transfer parameter (for system parameters only)
□	I	If exist	Transfer parameter if exists
□	N	Never	Never transfer parameter
del:	□	Identifier for deletion operations (i.e. parameter can be deleted). Not supported!	
table:	□	Table name in Agile e6	
field:	□	Table field in Agile e6	
format:	□	The format information for a transfer parameter is defined in the „format“ column (use „C“ programming language syntax). You need to make sure to use the appropriate data type for each parameter. Additionally, multiple text strings can be entered as default setting.	

Note The parameter marked with (*) are usually passed for items.

The following list contains examples of parameter definitions:

Value	Description
"y"	Fixed text entry – here with the default "YES"
"%s"	Undefined string
"%-20.20s"	Delimited string to max 20 chars, left justified
"% 10.2f"	Left justified 10-digit number with two comma positions.
"% 4.0f"	Left justified 4-digit number without comma positions.
"%8D"	8-digit date, (e.g. in "ECU.ini")

Each conversion rule is prefixed with the % character and has the general format:

```
%[<flags>][<field_width>].[<precision>][l]<conf_char>
```

The flag is usually represented by a “-“ character, i.e. the results o f a conversion procedure are displayed left-justified in the output field.

A user defined decimal character string defines the minimum field width. If the size in characters of the converted value is less than the field width it is padded, the missing leading characters are added at the left (padding) (or to the right if the "-" (left aligned) flag is enabled).

The precision parameter for the "f" conversion character in this example defines the number of places after the decimal point. The letter "l" for "long int" is not used here.

The conversion sign <conf_char> has the following function:

f	The float or double argument (arg) is converted to decimal display of the form "[-]ddd.ddd". The number of characters after the decimal point is defined by the precision value. 6 digits after the decimal point are assumed if no precision value has been defined. No decimal point is displayed if the specified precision is zero.
s	The argument (arg) is interpreted as a pointer to a character string. The output contains the number of characters specified by the precision value or the number of characters until the first \0 character. If no precision value has been defined, the output contains all characters until the first \0 character. The result is undefined if the argument value is NULL.

Transfer of Component Properties to PLM

The transfer of component properties to the document structure can be configured using the file docstr.par. If this file is missing, no component attributes are transferred to the PLM structure. There are several options for the configuration files *.par to transfer internal attributes (e.g. instance name, component name, reference set, position info, etc.) and component attributes. The following table describes the special options of docstr.par:

Option	Default Field	Description
S component.ident	T_DOC_STR.UG2_IDENT (Do not change.!)	CAX-Ident for this structure entry
S component.part	T_DOC_STR.CAX_COM (Do not 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 (Do not change.!)	Component reference set
S component.xmat	T_DOC_STR.ECC_XMAT (Do not 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 next

Option	Default Field	Description
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.

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

Header	Definition of create and update for a file 2006-06-30 // xPLM Solution						
	Mode A-Always (only system parameter) N-Never						
	S Standard A Attribut P &Parameter D Dimension V fixed value						
	type S Standard; V Default; E Environment						
	Typ	ident	mode	del	table	field	format
Table entries							
	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"
	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"

Example for alternate configuration in docstr.par for backward compatibility of assembly dependent auxiliary parts (transfers component name instead of part name into T_DOC_STR.CAX_COM). This does not work with enabled renaming or STEP options!

Header	Definition of create and update for a file 2006-06-30 // xPLM Solution						
	Mode A-Always (only system parameter) N-Never						
	S Standard A Attribut P &Parameter D Dimension V fixed value						
	type S Standard; V Default; E Environment						
	Type	ident	mode	del	table	field	format
Table entries							
	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"

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"

Transfer from Agile e6 to Unigraphics

Drawing Frames

Note In the mask EDB-DOC-HIS-RLI the default for the CAD_FLAG, which is called "Standard Entry-M", has always to be set to "Y". This is ensured if the loader file CadFlag.bld has been brought in the PLM system.

The integration uses "drawing tables" by default. Their content is based on the text field that is part of the drawing frame. When the function "Load frame" is used the drawing tables are filled with information from Agile e6.

The procedure is as follows:

1. The value of the parameter "cax_fra_ini" in the file ecu_ini.bat searches for a file containing the specification of the individual frame configuration files. By default, the name is ecu_fra.ini.

```
...
set cax_fra_ini=%ECU_ROOT%\ini\ecu_fra.ini
...
```

Each line of the file ecu_fra.ini refers to a possible frame configuration file. They are located in a directory defined by the setting of the parameter "pro_par_ini" in the file ecu_ini.bat .

The file ecu_fra.ini contains the following two lines by default:

1 ini/fra/standard.ini "Standard"	Identifies the standard text field with the

"Standard ECU V 3.0" EP DRAWING DRAWING	corresponding configuration file standard.ini.
2 ini/fra/ax.ini "Alternate" "Alternate ECU V 3.0" AX DRAWING DRAWING	Identifies the alternate text field with the corresponding configuration file ax.ini.

These entries refer to the example frames supplied with the integration. The lines always describe a configuration file and the storage location. Usually, it is the directory %ECU_ROOT%\ini\fra\.

The structure of each line is as follows:

- The sequential number of the entry is shown first.
- Followed by the path and name of each frame configuration file (entry relates to ECU_ROOT).
- Followed by the menu item and the pop-up text for the command "Change frame".
- The value "EP" or “AX” define the value of the DB-field T_DOC_DAT.CAX_SUBTYPE. This is used as a filter criteria in Agile e6.

2. Each Unigraphics frame has its own *.ini file. Frames are assigned to a configuration with a frame prefix.

When a frame is found in Unigraphics and a frame prefix is available, it is filled with information. The frame prefix defines the text to be stored in the file standard.ini.

```
...
# frame_prefix    = EP
 ...
```

The assignment of the entries from the configuration file standard.ini to the cells of the text field is made by entering the attributes in the corresponding frame. All Unigraphics functions for text formatting can be used.

Then the attribute is used to send identification information from Agile e6 to exactly this location. Except for entries in the history fields (starting field = history_prefix). Only the lines are passed depending on the offset stored in field "history_offset".

The format of the text comes from the text style of the attribute.

The entries of the standard configuration file standard.ini (see appendix) are based on the delivered example frames supplied with the integration.

standard.ini

The content of this standard configuration file is divided into several sections (described in the following table). The complete structure of the configuration files can be seen in the appendix.

Section	Parameter	Description
[Global]	part_type	Document type for parts.
	drawing_type	Document type for drawings.

Section	Parameter	Description
	start_field_of_history	Start field of the history entries in the text field.
	edb_mask	Information is read from this mask.
	update_frame	Frame processing when loading a drawing 0 - No frame is loaded 1 - If there is a frame in the drawing it will be loaded 2 - Frame is always loaded
	logiview_action	Post-action when loading (logic model / procedure).
	history_records	Number of displayed history entries .
	history_prefix	Prefix of the entity name for history fields (mark the starting field).
	history_data	Number of history fields in the history
	history_offset	Offset, number of counters before the next value appears in the history
	history_function_1	For internal use only.
	history_function_2	For internal use only.
	read_widget_mode	Widget mode for reading Form : Form name List : List name
	cad_flag_only	Always “y”
	file_manager	Always “1”
	permanent_load_list	1 - Use permanent load list 0 - Always generate new load list
	checkin_area	Not used
	frame_prefix	Identifies frame type
[Part]		Describes assignments of parts in Agile e6. No changes are allowed here!
[Document]		Describes the assignments of drawing information from Agile e6 to the drawing header in Unigraphics. The syntax for the Document section is as follows: ´table_name.field_name´ ´attribute ´ ´format´ The entries in the Document section determine the

Section	Parameter	Description
		master data from Agile e6 that is entered in the drawing. An attribute is a specific text block with the same name like the text field name that is predefined in the drawing frame. A zero (0) attribute has the effect that the assigned information from the document master data is not displayed in the text field. The 'format' entry can be used to influence the text format in the frame using syntax of the "C" programing language . The first part of the Document section should remain unchanged! Enter additional informationsin the second part, called "Optional Configuration Parameters". The last entry in this section may not be deleted !
[History]		This part also describes assignments of history entries in Agile e6 and in the UNIGRAPHICS drawing header. The syntax of the History section beginning with the history start field is as follows: ´table_name.field_name´ ´attribute ´ (no prefix) ´format´ The position of the history start field is defined by the parameter "history_prefix" in the "global section". It’s default attribute name is "ZVS100". In order to ensure that the function performs reliably the following parameters must not be deleted: T_DOC_HIS.FUNCTION T_DOC_HIS.CHANGE_STATUS T_DOC_HIS.HIST_ID

<table>
<tr><th>Section</th><th colspan="2">Parameter</th><th colspan="5">Description</th></tr>
<tr><td>[Special]</td><td colspan="2"></td><td colspan="5">The Special section defines special conditions for entering values from history of the document into special fields (e.g. user). The first value specifies the name of the field where a characteristic value is located.
The syntax for the Special section is as follows:
´table_name.field_name´ ‘value’
´table_name.field_name´ ´attribute´ flag ´format´
The "flag" refers to the occurrence of the information:
"1" most recent occurrence;
"0" first occurrence</td></tr>
<tr><td>[Total]</td><td colspan="2"></td><td colspan="5">This section allows to transfer additional data from Agile e6 into the history list. It is possible to define which data have to be transferred (e.g. the user and the date) depending of each particular status (e.g. "Released").</td></tr>
<tr><td></td><td>Function</td><td>Status</td><td>Flg</td><td>Fla</td><td>Field</td><td>Prefix</td><td>Format</td></tr>
<tr><td></td><td>Content of the EDB field T_DOC_HIS. FUNCTION</td><td>Content of the EDB field T_DOC_HIS. CHANGE_STATUS</td><td>flg</td><td>fla</td><td>Name of the table field where its content is entered as value of the frame attribute</td><td>´prefix´ of the frame attribute</td><td>´format´</td></tr>
<tr><td>[Equivalence]</td><td colspan="2"></td><td colspan="5">This part describes the assignment of material data from Agile e6 to the drawing header in Unigraphics.
The Equivalence section has two functions:
1. It controls the default for the fields of dependent EDB objects, items, models, and drawings.
2. It defines the display of EDB attributes in the drawing frames for specific models and items.
The table name for the entry "table_name" is mentioned in the first line.
The syntax for the Equivalence section is as follows:
´table_name.field_name´ ´table_name.field_name´
´table_name.field_name´ ´attribute´ ´format´</td></tr>
</table>

Using Custom Drawing Frames

The ECU software interface offers a number of predefined example drawing frames. However, most companies use either previously created custom drawing frames or create completely new custom frames. This document contains instructions on how to implement custom frames.

Texts use attributes to help you to format the text fields better.

1. Define a new standard form in the “Drawing Frames Mode” of Unigraphics and assign a unique name to it.
2. Once the format size, position, and unit have been defined, the outline of the new sheet is displayed on the screen.
3. Create a text field named "ZVSRAHMEN". This field has to fit into the previously defined frame and can be filled with text.
4. At the position where the text should be entered by the integration, a “pseudo-text” has to be placed in the desired format (e.g. font, size, and color).
5. Assign the text field to be filled by the integration with attributes and then assign them to the group "ZVSRAHMEN".
6. Save the new drawing format with a unique name in the corresponding standard directory of the integration (...ECU\frames).
7. We recommend to use a unique name with two characters (e.g. "AP"; P=private).
8. Create a parameter file with the same name by copying the standard file standard.ini and rename it (e.g. "AP.ini"). Store the file in the directory “…\ECU\ini\fra”.
9. Edit the file “ECU_fra.ini and add the following line:
10. ini/fra/AP.ini „Example“ “Example ECU V 3.5” AP DRAWING DRAWING
11. Edit the file AP.ini. Specify the attribute identifications of the fields to be filled out.

Chapter 6

Seed Parts and Drawings

Definition

Seed Parts determine the measurement basis for a new part. Uniform seed parts are used in a construction group for parts or drawings that are newly created.

The seed parts contain basic elements (planes, axes, coordinate systems), and predefined parameters.

Generally, a user can choose between custom specific seed parts or standard seed parts from the Agile e6 Unigraphics integration. The default object name and path is defined in the file ecu_ini.bat (or ECU_ini.ksh in UNIX).

The syntax for this set of variables in the file %ECU_ROOT%\ini\ecu_ini.bat is as follows:

```
set ecu_blank_part=%ecu_ini%\blank.prt
set ecu_blank_part_in=%ecu_ini%\blank_in.prt
set ecu_blank_drw=%ecu_ini%\drw.prt
set ecu_blank_drw_in=%ecu_ini%\drw_in.prt
```

Using Custom Specific Seed Objects

Seed parts can be easily customized. A drawing should be saved in drawing mode so that it is immediately reactivated when it is loaded (File – New). Enter the name and path of the file like in ECU_ini.bat.

Note Embedding of Manufacturing and Seedpart extensions is supported.

The usage of Manufacturing requires support of additional seed parts and triggers. When creating a new Manufacturing part, a currently loaded Unigraphics model is created inside. The next table shows the required environment settings that have to be performed in the file %ECU_ROOT%\ini\ecu_ini.bat (in MS Windows), respectively $ECU_ROOT/ini/ecu_ini (in Unix):

Environment Variable with Full Path	Subtype Environment (Standard)	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

The following table lists the called triggers for each subtype and the function number to be required in the file %ECU_ROOT%\ini\e6\ufcre.men (in MSWindows), respectively $ECU_ROOT/ini/e6/ufcre.men (in Unix).

Unit in mm

Subtype Unit	ufcre.men function number	TCL Pre-action Trigger	TCL Post-action Trigger
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 seed part	102	none	none
USER with ECU seed part	121	EcuCreateBySeedPart.tcl	none

Unit in inch

Subtype Unit	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 seed part	102 ❑	none	none
USER with ECU seed part	121	EcuCreateBySeedPart.tcl	none

In order to activate a new Create Dialog you have to place the following files in the ECU directories:

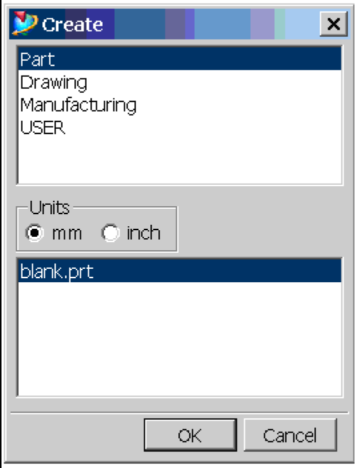
File	Destination Path	Description
ecu_cre.dlg	%ECU_ROOT%\application\%ECU_UGV%\application (in MS Windows). $ECU_ROOT/application/$ECU_UGV/application (in Unix)	UG dialog file
EcuCreateMenu.tcl	%ECU_ROOT%\tcl (in MS Windows) $ECU_ROOT/tcl (in Unix)	ECU dialog logic
EcuCreateMenu.ini	%ECU_ROOT%\tcl (in MS Windows) $ECU_ROOT/tcl (in Unix)	Seed part configuration file

The seed part configuration is performed in file %ECU_ROOT%\tcl\EcuCreateMenu.ini (in MS Windows), respectively $ECU_ROOT/tcl/EcuCreateMenu.ini (in Unix). The next table shows a sample configuration and the resulting Create dialog.

Sample Configuration in EcuCreateMenu.ini
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 #

Sample Configuration in EcuCreateMenu.ini

```
[ EcuCreateTypes ]
#
# Object type = CAX subtype
#
Part     =         NONE
Drawing    = DP
Manufacturing = MP
# Scenario   = SP
USER     = USER
#
```



Units ● mm ○ inch

Sample Configuration in EcuCreateMenu.ini

```
[ 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 ]
#
```

Type	Subtype	Description
[EcuCreateTypes]		Describes all possible seed part types. They are displayed in the type selection list of the dialog box and a CAX subtype is assigned to each of them.
	NONE	No CAX subtype is set, this means the base seed part and trigger is used
	DP	Drawing subtype. A drawing seed part and trigger is used

Type	Subtype	Description
	MP	Manufacturing subtype. A manufacturing seed part and trigger is used
	SP	Scenario subtype. A scenario seed part and trigger is used
	USER	The Part will be generated only in the local directory, not inside PLM. If no seed part is defined in [EcuSeedParts], the Unigraphics standard create userexit is called. If a seed part is defined in [EcuSeedParts], it can be selected in the dialog and the new part will be generated based on this seed part.
[EcuSeedParts]		Defines all seed parts. The dialog filters the selectable seed parts by the currently selected type and unit inside the dialog. The syntax of the seed part name (symbolic name on the left side) is: type_unit If there is more than one entry for the same symbolic name, the user can choose in dialog between more than one seed part. The internal ECU seed part file name is given on the right side. There are 3 options to create a seed part. The environment variable ecu_blank_part defines the path to the seed part. If seedparts in the [EcuSeedParts] do not contain the full path, the seed part must be created in the directory beside the part seed part. The internal ECU file name contains the full path. If the seed part file cannot be created by using the previous options, the dialog uses the old environment variables for seed parts, depending on the following subtype definitions. The environment variables contain the full path to the seed part.
	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

Type	Subtype	Description
	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
[DefaultSettings]		The default settings for unit or type of a seed part can be configured here.
	UNIT	Available values are: 0 - metric/mm (default) 1 - English/inches
	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 should not be modified. TYPE = 0 (default)

Embedding Configurable and Project Specific CAX Subtypes

Support of configurable CAX Subtypes requires modification of menu files and initialisation files. The TCL scripts EcuSetSubtype.tcl and EcuInit.tcl contain the logic and are required for execution of this menu item. It is recommendded to use always the latest version of that files. In the menu file agile.men / ecu.men an entry has to be added to replace all other menu items for setting and removing of subtypes :

```
BUTTON PLM_SET_SUBTYPE

LABEL Set

ACTIONS plm_action
```

The user is enabled to set project specific subtypes for the actual UG-object by executing the button “Set Subtype” The configuration of the available subtypes for this menu item is done in tcl/Ecu.ini.

The project specific subtypes have to be added to the respective sections in the file %ECU_ROOT%\tcl\Ecu.ini (in MS Windows), respectively $ECU_ROOT/tcl/Ecu.ini (in Unix) if not already existing. Below there is the delivered language specific configuration in this file:

```
[ 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
```

Chapter 7

CAX Plotting

Basics

CAX Plotting maintains plot jobs sent from any EDB client. (With Unigraphics NX3 the feature PQMGR is no longer available). To receive neutral formats of a plot, a file in CGM format needs to be created first which requires additional adaptations to generate additional formats (e.g. HPGL). For further information and requirements please contact Agile.

CAX Plotting involves triggering an immediate plotting process after checking the EDB vault or generating files in a neutral file format, such as TIFF, HPGL, CGM, or VRLM. It provides methods to integrate a wide variety of conversion programs in order to process jobs in batch mode at a central location (“batch client”).

The CAX Plotting converter with its license is required for only one computer. This computer maintains all conversion jobs that are sent from any client.

It converts from Unigraphics into the following data formats: JT, STEP, and DXF.

It can also be used to convert older Unigraphics documents into the current Unigraphics version, running as a background process.

The general process can be described as following:

- When a job is entered via CAX Plotting, an entry in the table T_CAX_PLT (the batch queue table) is made.
- A program running in the background (every 10 seconds) on the Agile e6 server checks for new entries in this table.
- When a new entry is found, the affected assembly is checked out completely and passed on to, e.g. the ugii2jt converter.
- JT files are generated for each component of the assembly and immediately checked back into the EDB vault.

Installation

Prerequisites

1. The conversion program is already installed (e.g. ugii2jt converter).
2. The required license is activated.
3. The conversion program must work correctly (this can be checked manually, e.g. with a part).
4. An EDB client must be installed on the respective computer.

Licensing

The following licenses are required for the batch tool CAX Plotting:

1. EDB license for normal client session.
2. Unigraphics open license for converters.
3. License for operation of the converter.
4. License for EAI Viewer to view resulting files.

Installation on Windows

To install the batch tool CAX Plotting (for ugii2jt converter) follow these steps:

1. Edit the file EcuBatchQueue.ini located in the CaxConfigDir and configure the variables for the runtime switch. (see EcuBatchQueue.ini).
2. Ensure that the converter works correctly with exclusive access. Otherwise, it will not work in batch mode. The correctness of the entries for the Converter Calls in the section EAIOptions in EcuBatchQueue.ini is important.
3. Edit the configuration file of the converter (by default tessUG.config). The name of this file must also be entered in the EAIOptions section in EcuBatchQueue.ini. (EcuBatchQueue.ini).

For more details about the converter refer to the online documentation.

Configuring the Data Model and the .ini-File

The entity CAX-PLT (table T_CAX_PLT) must exist in the EDB application of the batch tool "CAX Plotting". Another requirement is a working LogiView functionality in the logic model CxPlt. Both can be realized using the ASCII loader files CxPlt.dat and CxPltLGV.dat in Insert Mode ("I").

Note Regenerate the table after restarting the EDB client!

It is not necessary to load the file CxPlt.dat if the table T_CAX_PLT already exists. In this case only project specific fields need to be added manually.

A selection must be added to those Agile e6 menus from which the plot or conversion tasks should be generated:

```
Userexit: lgv_sel_run
```

```
Parameter CxPlt/CxPltECU
```

The LogiView procedure CxPlt/CxPltECU places an entry in the batch queue for the currently selected document. The string for the generating CAD system needs to be adjusted by editing the values for the parameter JobSystem in the file EcuBatchQueue.ini.

```
JobSystem= UGII
```

The LogiView procedure CxPlt/CxPltStpECU causes a Stop Job entry in the batch queue. The string for the generating CAD system also has to be adjusted by editing the LogiView procedure

(e.g. CxPltCreSys = "UGII")

EcuBatchQueue.ini

Note The only way to affect the behavior of this batch tool is to select the value for the appropriate switch in this file!

Section	Switch	Possible Value	Description
[Default]			
	EdbApplication	ugw_prod	Name of the running PLM application
	EdbUser	edbplot	Name of PLM user
	EdbHost	mchrme0a	Host computer where PLM server is running
	EdbStart	edb_start.bat	Startup script for the batch EDB client
	EdbStartRetry	10	Number of attempts to connect the client with the server
	EdbStartDelay	30000	Time between the start attempts in milliseconds
	EdbStop	True	Stops the PLM client and the batch tool
	JobMax	0	Number of jobs until batch tool stops automatically "0" = unlimited
	JobDelay	10000	Time in milliseconds until a new search is started for new entries in batch queue
	JobSystem	UGII	The batch tool only performs jobs with this search string as content of field T_CAX_PLT.CRE_SYSTEM
	JobDelete	False	Deletes entries of successfully completed jobs
	JobLogiView	NONE	Name of LogiView procedure which is performed before starting the plot
	QueueStopOn Exit	True	The batch tool is stopped when entering "exit"
	QueueStopOn File	C:\temp\stop.txt	The batch tool stops when this file exists

Section	Switch	Possible Value	Description
	QueueStopOnTime	23:00	The batch tool stops at the entered time "0" = never
	TempPlotDir	C:\plottemp\	Directory for various log files
	TempCheckOutDir	C:\plottemp\	Directory for checking out part files
	PQMGR_PlotDir	C:\hpgl	Directory to store *.hpp files, which are produced by the UG-PQMGR
	FileWaitTime	5000	Time in milliseconds until a new search of a file is started
	FileWaitRetry	50	Max number of trials to search a file
	FileWaitRetryUGExport	300	In general not used
	HpglToTiffConverter	/links/software/edb23/ ccu_test/tcl/hpgl2tiff.ksh	Directory and name of the start script for this converter
	CheckInCgm	True	Determines if the created .cgm file is checked in into the vault
	CheckInTiff	True	Determines if the created .tiff file is checked in into the vault
	CheckInHpgl	True	Determines if the created .hpgl file is checked in into the vault
	CheckInVrml	True	Determines if the created .vrml file is checked in into the vault
	FrameFill	ECU	Procedure name to complete a title block of a drawing
	FrameLoad	True	Procedure name to update a title block of a drawing
	OnlyActualDrawing	False	Generates only a plot of the current version of a drawing
	OnlyFirstDrawing	False	Generates only a plot of the oldest version of a drawing
	UpdateViews	False	Automatic update of “out of date” views

Section	Switch	Possible Value	Description
	BreakIfOutOfDate	False	Automatic break of “out of date” views or drawings
	SwitchView	True	Switches the version view using the parameters which where assigned to the plot order. Default is the release date of the document
	DeleteGroup	False	Name of group which is deleted before startup plotting. False = disabled
	DebugMessages	File	Name of file containing debug messages
	Quickload	False	Enables to use the quickload mode when set to "True"
[EAIOptions]			Enables to define options for the usage of the converter into the JT file format
[VDAOptions]			Enables to define the name and location of the configuration file for the usage of the VDA converter
[CGM]			Enables to qualify the output of a file in CGM format
	Pen1 (… 15)	1 (…. 15)	Mapping of plotter pen numbers
	PenSelection	1	1 = Thickness, 2 = Color
	TextRepresentation	2	1=Text, 2=Polylines
[Step214]			Enables the usage of the Step214 converter step214ug.exe The parameter STEP214UG_DIR hast to be set in file ecu_ini.bat
[DXF] and [DXFDWG]			Enables the usage of the DXFDWG converter dxfdwg.exe The parameter DXF_DIR and DXFDWG_DIR has to be set in file ecu_ini.bat
[IGES]			Enables the usage of the IGES converter iges.exe

Section	Switch	Possible Value	Description
[UGTO2D]			Enables the usage of the UG converter ugto2d.exe
[Parasolid]			Only Solids on this layer will be converted
[PrePlotAction]			Enables to add scripts here, which should be executed before the objects will be plotted or converted and the assembly is already loaded in Unigraphics
[PostPlotAction]			Enables to add scripts here, which should be executed after the objects are plotted or converted and the assembly is still loaded in Unigraphics
[ExportOptionsVRML]			Enables to define options for the usage of the converter into the VRML file format
[SelectQueueEntry]			Section is required to find the next plot queue entry
[DisplayQueueEntry]			Section is required to get necessary information about the actual plot queue entry from Agile e6
[UpdateQueueEntry]			Section is required to update the plotstatus of the actual plot queue entry
[InsertQueueEntry]			Section is required to insert a plot queue entry
[DeleteQueueEntry]			Section is required to delete a plot queue entry
[DisplayDocument]			Section is required to enable reading document information from document's master dataset.
[DisplayDocument_in_DocType]			Section is required to enable reading document information from document's type-specific dataset.
[DisplayDocDocLink]			Section is required to enable reading information about the document structure.

Section	Switch	Possible Value	Description
[DisplayDocFil]			Section is required to enable reading information about the file which belongs to the document.
[DisplayDocVerLink]			Section is required to display information about the version of the document.
[CheckInFile]			Section is required to get necessary information about the plot file to be saved in Agile e6
[CheckOutFile]			Section is required to get necessary information about the file to be loaded from Agile e6
[StatusInfo]			Section is required to get release status information from Agile e6
[InsertHistory]			Section is required to get history information from Agile e6

Tips and Tricks

Large Assemblies

Add Component from PLM

For large assemblies the regeneration of the display can take a long time after the command “Add component” has been called. But a component can be added hidden (i.e. in silent mode) and therefore, no change of the display is necessary.

In the file ufrcp.men the entry 120 refers to the hidden open component function.

```
PLM
PLM
Help text
Help text
120
```

Suppress Loading of WAVE Parents During Save

Loading WAVE parents can take some time for large assemblies. This can be suppressed by changing the environment switch in the [Initialize] section of the tcl/Ecu.ini:

```
EcuWaveLoadParents = 0
```

Note When setting the value to 0 the adapter cannot detect the link source of unloaded WAVE parents.
Use ANT to make sure all required WAVE parents are loaded if the WAVE link information should be stored into PLM

Updated Unigraphics Version

When the Unigraphics version has changed, it has to be updated in Agile e6 as well.

Edit the line "cre_sys" in the sections [Create] and [Update] of the parameter files *.par and *.min. This will transfer the UG NX version number of the saving UG session.

```
V    cre_sys    A    N    T_DOC_DAT    CAX_CRE_SYSTEM    "UG-%s"
```

Chapter 9

Debugging

The most profound view into the integration’s functioning enables tracing the ECI-calls which are sent from the local Agile client to the connected Agile server including the responds. A file is created on the Agile server machine which can be viewed by user using the client.

It is possible to start and to stop the ECI-trace at any time during the session.

The trace function is activated in Agile e6:

1. Click Tools > Trace > Select Module
2. Select the checkbox for ECI-Module E 9.
3. Click OK.
4. To start the trace log any time during the session, click Tools > Trace > Trace New .

   The name of the new trace file is displayed in the message bar (e.g. Test output on D:\AgilePLM\tmp\tst173.trc).

Note The file location is that on the Agile server machine!

5. Execute the process you want to test/trace.
6. To end tracing, click Tools > Trace > Trace Off.
7. To display the contents of the trace file in a list, click Tools > Trace > Show trace.

   You can copy and paste the trace contents into an Excel file and save it locally.

The information in the trace file can be extended by e.g. adding SQL statements.

Chapter 10

Appendix

Content of \tcl\Ecu.ini

Note To guaranty a correct behavior of the integration during initialization, values for the respective switches have to be selected in the file \tcl\ecu.ini.

Note According to the syntax of Tcl programming language the usage of The sign “#“ in front of the switch name means that this switch is not used. Do not remove this sign!

Section	Switch	Value	Sphere	Description
[Initialize]				
	DialogAttributes	EDB_NAME	Save menu	Value can be a comma separated list of names, parts, attributes, which are also displayed in the dialogs of the ECU integration
	EcuCustomerID	NONE	General	CustomerID used for customizing scripts
	EcuDebugLevel	0	General	Specifies which debug mode is possible inside the integration (0=not set, 1=set)
	EcuStylerDialog	1	Dialog	Full functionality of dialogs on UG-site (starting with integration version 3.6)
	EcuBookmarks	1	Load menu Save menu	Activated = UG bookmarks will be saved in the fileserver Deactivate = converts the row into a comment, preceding with a “#”
	Wave	1	Load menu Save menu	Analyzes “wave links” and saves them in PLM; this option can be manually deactivated in the “save” dialog
	EcuWaveLoadParents	1	Load menu	Enables to suppress loading of "wave" parents. 0 = the integration cannot detect the link source of unloaded "wave" parents. Use

Section	Switch	Value	Sphere	Description
				ANT to make sure all required "wave" parents are loaded, if the "wave" link information should be stored in PLM.
	CaxNoReserve	1	Save menu	To increase the performance during save operations, the feature “automatic reservation” is deactivated if the value is set to “1”
	EcuTimestamp	1	Load menu Save menu	Enables usage of “timestamp” function to observe model modifications and conflicts at concurrent engineering phases
	EcuUseCidFile	1	Load menu Save menu	Enables buffering of ID data in several files of the type .ecu; EcuTimestamp has to be activated
#	EcuRename	1	Load menu Save menu	Enables the “rename” function; old documents may have to be updated in order to adjust the interrelationship between the parameter files of the ecu-integration and the document data and document structure
#	EcuStepTransformation	1	Load menu Save menu	Enables modeling by using the transformation matrix
#	EcuWriHisMod	1	Save menu	To increase the performance during the saving process, the value of the switch “EcuTimestamp” is set to 1 so no documents history CAX CHANGED entry is created. If the value for “EcuWriHisMod" is set to “1” a documents history CAX CHANGED entry is created..
#	Save Menu appearance			Default: 0 = Not set; 1 = Set
	EcuUISaveWaveValue	1	Save menu	Activates the default wave.
	EcuUISaveWaveVisible	1	Save menu	Activates the visibility of wave.

Section	Switch	Value	Sphere	Description
	EcuUISaveWaveEnable	1	Save menu	Activates the modification of wave.
	EcuUISaveFreeValue	1	Save menu	Activates a default of the ”dereservation”
	EcuUISaveFreeVisible	1	Save menu	Activates the visibility of the ”dereservation”
	EcuUISaveFreeEnable	1	Save menu	Activates the modification of the ”dereservation”
	EcuUISaveOptionDefault	1	Save menu	0 = Interactive; 1 = Batch; 2 = User; 3 = Quick
	EcuUISaveDisplayDefault	1	Save menu	0 = Tree; 1 = List
	EcuUISaveUser	1	Save menu	Activates the saving option “user”.
	EcuUISaveQuick	1		Activates the saving option “Quick”.
	EcuUISaveAsOptionDefault	0	Save menu	0 = Interactive; 1 = Batch 2 = User; 3 = Quick
	EcuUISaveAsDisplayDefault	0	Save menu	0 = Tree; 1 = List
	EcuUISaveAsUser	1	Save menu	Enables saving as user 0 = Not allowed; 1 = Allowed
	AbortSaveOnFailure	1	Save menu	When activated the saving process is stopped completely when a failure occurs; a roll-back does not happen in the vault
	EcuCheckStatusOnSave	1	Save menu	When activated the PLM status is checked before rising up the save dialog. otherwise checking the PLM status is done after pressing the OK button
	EcuNeverLoadComponentOnSave	0	Save menu	When activated the subcomponents of user assemblies are not loaded

Section	Switch	Value	Sphere	Description
	EcuRegenerateAfterSave	0	Save menu	When activated the display of a drawing will be regenerated after saving the drawing, required for workaround a NX display refresh bug
#	Load Menu appearance			in most cases 0 = Not set; 1 = Set
	EcuUIOpenUser	1	Load menu	Enable to open from a directory
	EcuUIQuickload	1	Load menu	Enables button to activate Quickload
	EcuUIAttributes	1	Load menu	Updates attributes during load process
#	ecu_multiple_article	multiple	Frames	This string is written into the title box if multiple articles are assigned to one drawing
#	EcuFrameInPdm	1	Frames	Frames and their initializations are loaded and saved from the PLM database but NOT from the frame directory (currently not supported)
#	EcuDocumentQuantity	1		Transfer of the component count into the document structure

Content \tcl\EcuQuickLoad.ini

Note To guaranty a correct behavior of the integration during initialization, values for the respective switches have to be selected in the file EcuQuickLoad.ini.

Note According to the syntax of Tcl programming language the usage of The sign “#“ in front of the switch name means that this switch is not used. Do not remove this sign!

Section	Switch	Value	Sphere	Description
[Init]				
	Quickload	1	Load menu	1 = PLM mode; 2 = SMT mode 3 = QuickLoad deactivated, normal (=slow) ECI load happens“

Section	Switch	Valu e	Sphere	Description
	QuickloadAttributes	0	Load menu	0 = No updates of attributes; 1 = Attributes are updated Value 1 is not recommended. All fields must be defined in the type mask (mode 3-ECI) and in the relation mask (mode 2-SMT) Before performing the ECI call the fields are automatically added like defined in the section EcuGetDocumentInfo in the file ecu.ini. If Quickload=1, this switch is ignored because of technical reasons.
	QuickloadConfirm	0	Load menu	Controls the behavior when loading 0 = Parts are loaded without query 1 = If a version conflict is detected, user has to decide 2 = Parts are replaced without query
	QuickloadPackage	30	Load menu	Defines the internal package size for “standard PLM” quickload
	QuickloadCkoAll	0	Load menu	Activates the checkout of all files 0 = Deactivated; 1 - Activated
	QuickloadCkoAllAsk	0	Load menu	0 = Selective ECI checkout triggered by the ECU integration 1 = Standard PLM (checkout all files) 2 = User is asked before each checkout of a file
	QuickloadCkoFms		Load menu	If QuickLoad is activated: 0 = Standard ECI checkout 1 = Unsupported FMS checkout 2 = DFM checkout (works only if QuickLoad is set to “1” or “2” on NT)
	QuickloadDebug	0	Load menu	0 = No debug info 1 = Debug info in a separate UG window

Section	Switch	Valu e	Sphere	Description
	QuickloadMessage	1	Load menu	0 = No messages while loading 1 = Shows load info in a UG window
	QuickloadNotGlobal	0	Load menu	0 = Do not cancel loading in a global view 1 = Cancel load in global view
	StepLayer	0	Load menu	Unsupported, for future features.
	QuickloadCheckAccess	0	Load menu	If QuickLoad is activated: 0 = Do not check 1 = Check access and view
	QuickloadOnlineInBatch	1	Load menu	1 = Online replication, but requires changes in DFM procedures 0 = Cancel
	QuickloadUpdateFrame	0	Load menu	1 - After load, frame is updated, 0 – After load, frame is not updated, 2 - After load user is asked for update
	LogInfoOnLevelIndex	0	Load menu	Comma separated list of level index that should be displayed after load

Content of \tcl\EcuMigrate.ini

Section	Switch	Value	Sphere	Description
[Init]				
	EcuUseUGPC	0	Migration	Obsolete
	LogErrors	1	Migration	Enables writing a log file 0 = Not set; 1 = Set

Section	Switch	Value	Sphere	Description
	EcuSearchById	1	Migration	1 = Search 2 = Create a new data set If “1”, a PLM data set is searched using the attributes which were defined in [FindAndUpdateById] UG information and file information is added to this data set.
	MapFile	D:\projects\ecu\develop\tcl\export.att	Migration	Path and name of an attribute mapping file. Its internal structure format agrees with the file …\tcl\ecu.ini [partname.prt].
	NormPartLocal	0	Migration	Enables local storing of norm parts 0 = Check-in; 1 = Store local
	History	20	Migration	Number counter for SF_* text attributes which are to be read (e.g. this can correspond with the maximal number of rows in a table of a drawing
	OnlyOnePartPerDrawing	0	Migration	Important when checking in master model drawings. Enables adjusting to one or more models per drawing 0 = A drawing with more than one 1st-level model reference is allowed 1 = A drawing with more than one 1st-level model reference is treated as an assembly
	RenameOnImport	0	Migration	Enables renaming during the import procedure 0 = Not set; 1 = Set