

Agile
Version e6.0

ORACLE®

Oracle Agile Engineering Data Management

Oracle Agile Engineering Data Management - MCAD
Connector for Pro/Engineer – Version 3.7.2.0
Installation and Administration Manual

Part No. E12425-01
April 2008

Copyright and Trademarks

Copyright © 1995, 2008, Oracle. All rights reserved.

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

Copyright and Trademarks	ii
Introduction	7
Overview	7
Architecture	7
Installation	9
Prerequisites	9
Checking the Prerequisites	9
Unpacking the Archive File	9
Adapting the Startup Script	11
Special Customizing for Windows XP 64bit	12
Customizing Vault Options	13
Configuration	15
General Concept	15
Configuring the User Interface	15
Overview of the Initialization Files	16
Menu Structure	21
The Menu Structure	21
Settings in EcpCustomer.ini	21
Structure of the Menu Text	22
Harmonization of the Menu Structure	23
Parameter Transfer	25
Transfer from Pro/Engineer to Agile e6	25
Description of Parameter Entries	26
Support of TCL Inside ECP	36
Transfer from Agile e6 to Pro/Engineer	37
Transfer of PLM Data into Frames (Title Block) of Drawings	39
Structure of Standard.ini	41
Using Custom Drawing Frames	44
Using LogiView	45

I18N and L10N	46
I18N Basic Support	46
L10N Software Design Features	47
Debugging.....	48
Appendix	50
Including Your Own Menu Entries / Menus.....	50
Including Your Own Submenus.....	51
Overview of ECP_Master.ini Settings	52

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) (<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) (<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) (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.

Introduction

Overview

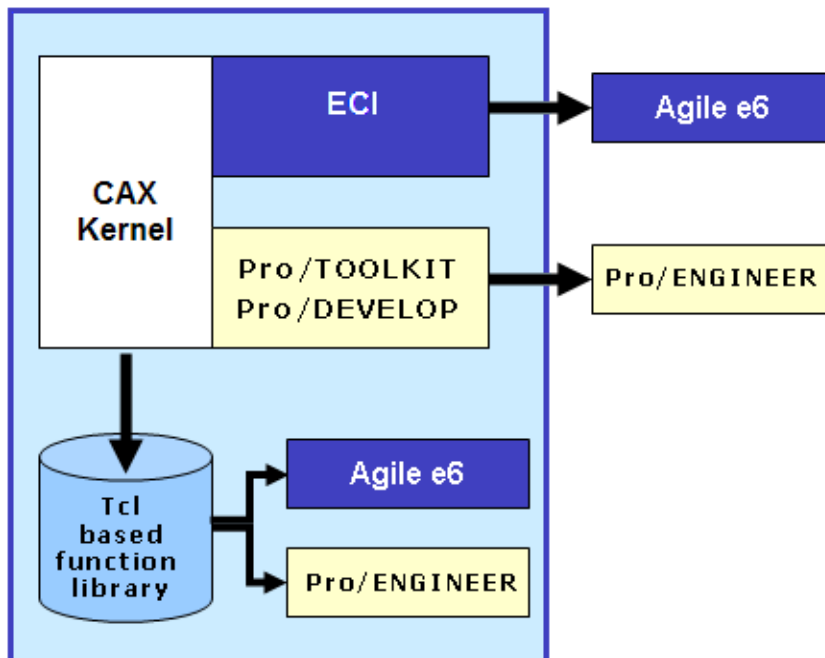
The information in this document is based on a standard installation and the actual appearance of a specific installation may vary depending on the local environment and custom configuration realized for the individual requirements of the customer.

The integration functions are provided with additional and expanded menus in Agile e6 and Pro/Engineer which access the ECI (External Communication Interface) for CAD and Agile PLM system functions and the CAD and Agile PLM. All management data are entered interactively in Agile e6 allowing a variety of document types.

Architecture

The Agile e6 Pro/Engineer integration was created using the development tools Pro/TOOLKIT and Pro/DEVELOP. It represents an enhancement to the operations of Pro/Engineer. The Agile e6 Pro/Engineer integration (ECP) is an **add-on module** to Agile e6.

The architecture of the integration between Agile e6 and Pro/Engineer is illustrated in the following graphic:



The integration can be customized according to customer needs by editing an analog file with the string “custom” in its name, e.g. edit “EcpCustomer.ini” instead of “EcpMaster.ini”.

Custom configurations can be realized matching the requirements of an individual situation. This is the content of Chapter 3.

We offer for example a uniform GUI which can operate with different operating systems throughout the company. This facilitates the operation of the integration despite the use of different operating systems.

Installation

Note After a correct installation, a basic structure of the integration is running. The original files, especially the text files containing default values should NOT be changed.

The installation of the Agile e6 Pro/Engineer integration ECP is divided into several steps. Performing the steps in the proper order is important for correct operation.

The following steps must be carried out:

1. Prerequisites
2. Checking the Prerequisites
3. Unpacking the Archive File
4. Adapting the Startup Script
5. Special Customizing for Windows XP 64bit
6. Customizing Vault Options

Prerequisites

The Agile e6-Pro/Engineer integration is currently available for several hardware platforms and operating systems. Please see the Release Notes for an overview.

Checking the Prerequisites

- Agile e6 is installed and running.
- The Agile e6 file management service (FMS or DFM) works properly.
- A designated user environment is already set up in Agile e6.
- In Agile e6 a test user with a valid password exists and is authorized to start the Agile e6 session.
- A license for using the ECP integration is registered and available on the FELICS license server (you can obtain the ECP licenses from the Agile sales representative).
- Pro/Engineer is installed and can be started from the HOME directory.
- The integration works with the Windows XP 64bit operating system provided Wildfire2 M180 or higher and Agile e6.0.2 or higher are used.

Unpacking the Archive File

To install the ECP Integration extract the delivered zip file to any path on your hard drive (e.g.

C:\AgilePLM\ecp37)

The installation creates the following directory structure:

Main Directory	Sub-directory	Sub-directory	Description
\addons			Contains additional parts of the ECP integration. Special customizing needs to be performed here.
	\Baseline		Contains software and detailed documentation for activating and using this feature.
	\EcpPlt		Contains loader files to establish the plot management.
	\I18N		Contains files to realize basic support of internationalization.
	\Menu		Contains example files to include user specific menu expansions.
	\Mfg		Contains script files to manage NC data files.
\com			Contains a start script for the Agile e6 client.
\ecpusr			Contains user specific data. Every user has the same separate subdirectory tree. After the installation it is created for the administrator only. Several lists (status, object list, etc.) reside in this directory tree. All directory definitions for an individual, user-specific directory can be modified.
	\engineer_A		Contains individual user specific data for username "engineer_A".
		1. \backup 2. \temp 3. \trail 4. \work	1. Contains files that could not be saved at the intended location. 2. Used as temporary storage directory for file server operations for this user. 3. Contains trail files for individual Pro/Engineer sessions. 4. Used as local working directory of this user.
\english.*			There are two parallel language dependent trees, \englishplm and \english.e6, containing the text resources (for e6 and for the older plm versions)
	\text		Contains message text and menu text in English.
		\menus	Contains auxiliary menu extension files (*.aux) and old menu files in English (*.mnu) if still used.

Main Directory	Sub-directory	Sub-directory	Description
		\resource	Contains resource files for the dialog in English (*.res), and bitmap files (*.gif).
\german.*			There are two parallel language dependent trees, \german.plm and \german.e6, containing the text resources (for e6 and for the older plm versions)
	\text		Contains message text and menu text in German.
		\menus	Contains auxiliary menu extension files (*.aux) and old menu files in German (*.mnu) if still used.
		\resource	Contains resource files for the dialog in German (*.res), and bitmap files (*.gif).
\frames			Contains examples of drawing frames and tables.
\i486_nt			Contains platform specific libraries (for Windows NT only). The file name reveals the specific usage (e.g. ecp_e602_2003.dll is the binary to connect Agile e6.0.2 and pro/Engineer Wildfire2).
\ini			Contains initialization files.
	\fra		Contains drawing frames.
	\par		Contains parameter files.
\tcl			Contains TCL support scripts.
	\cust		Contains the TCL script EcpCust.tcl as an example for locating TCL scripts. Such scripts are used to enlarge the PLM menu structure by custom specific submenus.
\tcllib			Contains TCL support libraries
\x86e_win64			Contains all dll libraries to support MS Windows XP64

Adapting the Startup Script

To set the customer specific variables the respective startup file in the ECP_ROOT_DIR needs to be adapted:

- ECP_START.BAT (MS-WINDOWS)
- ecp_start.ksh (UNIX)

This startup file calls scripts to start a PLM client as well as a Pro/Engineer session.

The settings are specific to your current installation. After adapting the following variables the ECP integration should already work.

Variable	Description	Valid content (default setting)
start_pro	Full name of the actual command procedure to startup Pro/Engineer	c:\Program Files\proeWildfire\bin\proewildfire.bat
EDB_DIR	Location of the Agile e6 software	c:\Program Files\AgilePLM
ECP_ROOT_DIS	Disc location of the integration ECP	C:
ECP_ROOT_DIR	Directory path of the integration ECP	\Program Files\AgilePLM\ecp37
ECP_PEV	A number representing the version of the software product Pro/Engineer	2006
ECP_CV	A number representing the software product Agile e6	602
LANG	Menu language of the Pro/Engineer system	English / German
ECP_LANG	Menu language of the software	English / German
ECP_USR_DIS	Drive letter of the location of the user's private directory	%ECP_ROOT_DIS%
ECP_USR_PATH	Path to the location of the private directory of the user	%ECP_ROOT_DIR%\ecpusr\%USERNAME%
ECP_USR	Complete name of the location of the private directory of the user	%ECP_USR_DIS%%ECP_USR_PATH%
USRNAM	Agile e6 user definition	Will be transferred at startup of a client session. Usually identical to the OS user.

Special Customizing for Windows XP 64bit

The given ECP directory structure has to contain a new directory which contains all dll libraries to support Pro/Engineer and Agile e6.0.x running on Windows XP64.

Please verify that the directory “x86e_win_64” is located in the ECP_ROOT_Directory.

Modify the values in “ecp_start.bat” and “ecpplt.bat” according to your installation.

Example “ecp_start.bat”:

```
rem *****
rem Settings (Modify by installation)
rem *****
rem [comment] ==> Pro/E start command. Make shure to use .bat instead of .exe
set start_pro=C:\Program Files\proeWildfire 3.0\bin\proe1.bat
rem
rem [comment] ==> Physical location of of Agile e-Series Client
set EDB_DIR=D:\Agile_e6_Client
rem
rem [comment] ==> Partition where ECP Connector is installed
set ECP_ROOT_DIS=D:
rem
rem [comment] ==> Path to ECP_ROOT_Directory in the above given Partition
set ECP_ROOT_DIR=\ProENGINEER_ecp3710\ecp37
rem
rem [comment] ==> As a default Wildfire3 will start the ECP_JAVA_Client (ECP_PEV=2006)
rem [comment] ==> Use (ECP_PEV=2003) to start Wildfire3 with ECP_Windows_Client
rem Pro/E Version (2001 -> 2001; Wildfire -> 2002; Wildfire2 -> 2003; Wildfire3 -> 2006)
set ECP_PEV=2006
rem
rem [comment] ==> select the appropriate Agile Software Version.
rem EDB Version ( 50 -> plm 5.x; 60 -> Agile e6; 602 -> Agile e6.0.2 and higher)
set ECP_CV=602
```

Customizing Vault Options

The default name of the used vault is “PROE” (capitalized). If you want to use another name, you need to change it in Agile e6.

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

Configuration

General Concept

If the integration is set up correctly, it works in default mode - the content of the file ...\\ini\\EcpMaster.ini triggers the behavior. You can now configure the behavior of your integration.

Note The content of the file EcpMaster.ini must NOT be changed at all!
 Always edit the file ...\\ini\\EcpCustomer.ini !

Note Each file *.ini must not contain blank lines! Use a trailing # sign for a comment line.

To configure your integration, overwrite the default trigger value of a particular variable in the customer specific file EcpCustomer.ini. If you want to use another file name, it has to be stored as value of the variable EcpCustomerIni first which is located in the file ecp_start.bat.

It is possible to get a summary about the current settings by changing into the debug mode and saving the output in a so-called trace file.

1. Activate the debug mode in Pro/E via PLM > Tools > Debug > Batch.
2. Select PLM > Tools > Reload Config to get the summary.
3. To close the trace file, select PLM > Tools > Debug > Close.

The location and the name of the trace file are displayed on the Pro/E message bar (e.g. set_deb_001: trace file: D:\\CAD\\ecp37\\ecpusr\\agile\\tmp\\cax_1992.trc closed).

Optional, it is possible to extend the generated ECI error information by adding the following line to the file EcpCustomer.ini:

```
EcpGetAddErrorInfo = 1
```

Configuring the User Interface

The integration uses the following default masks that can be configured:

EDB-ART-DOC-RLI	EDB-ART-DOC-ALI	EDB-ART-DOC-ALI-C
EDB-ART-SFR	EDB-ART-SLI	EDB-ART-STR-RLI
EDB-BAS-USER-VLI		
EDB-DOC-CFR	EDB-DOC-SFR	EDB-DOC-SLI
EDB-DOC-FIL-RLI-C		
EDB-DOC-STR-RLI	EDB-DOC-STR-ALI	

EDB-GRP-CFR

EDB-GRP-SLI

EDB-GRP-ART-RLI

Overview of the Initialization Files

Initialization File	Description
EcpDocument.ini	Defines various object types that are used throughout the software (for further information see next table). If you want to make changes it is recommended to use another name for this file. Remember to change the value of the parameter EcpDocumentIni in the file EcpCustomer.ini (Default: "EcpDocument.ini") as well.
EcpTools.ini	The content of this file must not be changed at all
CaxGtm.ini	The content of this file must not be changed at all
ecp_fra.ini	The content of this file must not be changed at all
EcpBatch.ini	The content of this file must not be changed at all
EcpEAI_Full.ini	The content of this file must not be changed at all
EcpParameter.ini	The content of this file must not be changed at all
Igv.ini	The content of this file must not be changed at all

Main sections of the file ecpDocument.ini:

Section	Description
Cax Type	Entries in this file represent an assignment between CAX object type and document type. An Agile e6 document type is assigned via the file extension of the corresponding CAD file. You can modify the assignment configuration in the corresponding "cax type" line.
Document Type	Entries in this section define assignment references in Agile e6.

[Cax Type] Section Parameter

Parameter	Value	Description
Part	PRT	Assigned to 3D models.
Assembly	ASM	
Layout	LAY	
Manufacturing	MFG	
Drawing	DWG	Assigned to Drawing.

[Document Type] Section Parameter

Parameter		Description						
Entity		Used entity in Agile e6						
Type		Used document type in Agile e6						
	Mask	<p>Defines the mask to be displayed in Agile e6 for the corresponding file type.</p> <p>Syntax example:</p> <table> <tr> <td>EDB-DOC-DRW-TLI</td><td>EDB-DOC-DRW-TFR</td><td>T_DOC_DRW</td></tr> <tr> <td><u>List</u> to be used for load operations</td><td><u>Form</u> to be used for save operations</td><td><u>Table</u> containing the data of the corresponding document type</td></tr> </table>	EDB-DOC-DRW-TLI	EDB-DOC-DRW-TFR	T_DOC_DRW	<u>List</u> to be used for load operations	<u>Form</u> to be used for save operations	<u>Table</u> containing the data of the corresponding document type
EDB-DOC-DRW-TLI	EDB-DOC-DRW-TFR	T_DOC_DRW						
<u>List</u> to be used for load operations	<u>Form</u> to be used for save operations	<u>Table</u> containing the data of the corresponding document type						
	Widget	<p>Defines the appearance of the displayed mask for save operations in Agile e6. The following options are available:</p> <p>Form form view</p> <p>List list view</p> <p>bvb opens form using BVB userexits</p>						

Parameter		Description																														
	Fields	<p>Contains Agile e6 fields which must be available to perform data transfer.</p> <p>The sequence and number of the entries in the "mask" line and in the "fields" sections must not be changed at all!</p> <p>Explanation:</p> <table><tr><td>T_DOC_DAT.C_ID</td><td>Unique Identifier</td></tr><tr><td>T_DOC_DAT.DOCUMENT_ID</td><td>Document Number</td></tr><tr><td>T_DOC_DAT.SHEET_NO</td><td>Sheet Number</td></tr><tr><td>T_DOC_DAT.CAX_CRE_SYSTEM</td><td>Abbreviation describing the Creating System</td></tr><tr><td>T_DOC_DAT.CAX_TIMESTAMP</td><td>Timestamp specific to the integration's work</td></tr><tr><td>T_DOC_DAT.DOC_TYPE</td><td>Document Type</td></tr><tr><td>T_DOC_DAT.CAX_VAR</td><td>Name of the Instance</td></tr><tr><td>T_DOC_DAT.CAX_TYPE</td><td>Object Type in Pro/E</td></tr><tr><td>T_DOC_DAT.CAX_SUBTYPE</td><td>Sub-type</td></tr><tr><td>T_DOC_DAT.CAX_FRAME_ID</td><td>ID of the used frame</td></tr><tr><td>T_DOC_DAT.CAX_STROK</td><td>Structure Data Flag</td></tr><tr><td>T_DOC_DAT.CAX_LOCAL</td><td>Load Location Flag</td></tr><tr><td>T_DOC_DAT.CAX_FIL_DISC</td><td>File location - Disc</td></tr><tr><td>T_DOC_DAT.CAX_FIL_PATH</td><td>File location - Path</td></tr><tr><td>T_DOC_DAT.CAX_FIL_NAME</td><td>File Name</td></tr></table>	T_DOC_DAT.C_ID	Unique Identifier	T_DOC_DAT.DOCUMENT_ID	Document Number	T_DOC_DAT.SHEET_NO	Sheet Number	T_DOC_DAT.CAX_CRE_SYSTEM	Abbreviation describing the Creating System	T_DOC_DAT.CAX_TIMESTAMP	Timestamp specific to the integration's work	T_DOC_DAT.DOC_TYPE	Document Type	T_DOC_DAT.CAX_VAR	Name of the Instance	T_DOC_DAT.CAX_TYPE	Object Type in Pro/E	T_DOC_DAT.CAX_SUBTYPE	Sub-type	T_DOC_DAT.CAX_FRAME_ID	ID of the used frame	T_DOC_DAT.CAX_STROK	Structure Data Flag	T_DOC_DAT.CAX_LOCAL	Load Location Flag	T_DOC_DAT.CAX_FIL_DISC	File location - Disc	T_DOC_DAT.CAX_FIL_PATH	File location - Path	T_DOC_DAT.CAX_FIL_NAME	File Name
T_DOC_DAT.C_ID	Unique Identifier																															
T_DOC_DAT.DOCUMENT_ID	Document Number																															
T_DOC_DAT.SHEET_NO	Sheet Number																															
T_DOC_DAT.CAX_CRE_SYSTEM	Abbreviation describing the Creating System																															
T_DOC_DAT.CAX_TIMESTAMP	Timestamp specific to the integration's work																															
T_DOC_DAT.DOC_TYPE	Document Type																															
T_DOC_DAT.CAX_VAR	Name of the Instance																															
T_DOC_DAT.CAX_TYPE	Object Type in Pro/E																															
T_DOC_DAT.CAX_SUBTYPE	Sub-type																															
T_DOC_DAT.CAX_FRAME_ID	ID of the used frame																															
T_DOC_DAT.CAX_STROK	Structure Data Flag																															
T_DOC_DAT.CAX_LOCAL	Load Location Flag																															
T_DOC_DAT.CAX_FIL_DISC	File location - Disc																															
T_DOC_DAT.CAX_FIL_PATH	File location - Path																															
T_DOC_DAT.CAX_FIL_NAME	File Name																															
	file_name	<p>Defines the method how to generate the file names.</p> <p>The following methods are available:</p> <p>(%id) Document ID is used to generate the filename</p> <p>(%sh) Sheet number is used to generate the filename</p> <p>Generally, it is possible to use other sources too. This is extended customizing. You must write a mask trigger working at INSERT-case. This trigger has to compose the file name and write it into the mask field T_DOC_DAT.CAX_FIL_NAME . The maximum length is limited by the maximum possible length in your Pro/Engineer version.</p>																														

Parameter		Description
	checkin_area	<p>Defines the corresponding security area of the file server. Use the correct syntax (entries are case sensitive!) Example for a UNIX environment:</p> <p>checkin_area PROE</p> <p>If "Distributed File Management" (DFM) is used, the checkin_area entry must be <i>CaxUseDefault</i>.</p>

Menu Structure

The Menu Structure

The menu structure is defined in the `ecpmain.txt` file. The `ecpmain.txt` file can be found in the respective text directory of the used software version (e.g. `...\English.e6\text\ecpmain.txt`). If you want to add your own structure it is recommended to copy and rename the file `ecpmain.txt` and apply your changes in here. The reference to this file has to be set in the `EcpCustomer.ini` in the line `EcpMenuMainRes`.

Note In order to facilitate menu extensions, there is a complete example already available in the directory `..ecp36\addons\Menu`.

Settings in EcpCustomer.ini

In order to use the renamed file simply replace the line in section `[Initialize]` of the file `EcpCustomer.ini`:

```
EcpMenuMainRes = ecpmain.txt
```

By:

```
EcpMenuMainRes = <renamed copy>.
```

`[EcpMenu]` is a separate optional section in `EcpCustomer.ini` for mapping user defined menu entries for the used TCL scripts. A subdirectory of the TCL directory can be included in the definition.

Syntax:

```
<internal menu name> = <script name>.tcl
```

or

```
<directory>\<script name>.tcl
```

Example:

```
#
[ EcpMenu ]
#
EcpMainMenuEntry100      = cust\EcpCust1.tcl
EcpCustMenuEntry1       = cust\EcpCust2.tcl
```

Structure of the Menu Text

Note Text must be located in the Text directory of the ECP (must comply with the definition in protk.dat).

There are always 8 lines in a block. The first block always contains the menu definition itself. The definitions for menu entries and submenus start in the 2nd block.

To convert a menu entry to a comment, add a # in front of the lines 1, 2, 5 and 6.

Note Menu entries and names must be unique (within Pro/E).

Block Structure of a Menu Entry

Line	Content	Description	Values								
1	ECPMAIN_Entry100	<p>The internal Pro/E name of the menu entry (label) is generated here. It must start with ECP and end with _Entry<number> (except for the 1st block).</p> <p>Syntax of internal Pro/E name of the menu entry: <menu name> Entry<number>.</p>	<p>Numbers:</p> <table><tr><td>up to 39</td><td>Internal menu entries;</td></tr><tr><td>40-49</td><td>TCL functions;</td></tr><tr><td>50-99</td><td>new TCL functions;</td></tr><tr><td>≥100</td><td>user entries</td></tr></table>	up to 39	Internal menu entries;	40-49	TCL functions;	50-99	new TCL functions;	≥100	user entries
up to 39	Internal menu entries;										
40-49	TCL functions;										
50-99	new TCL functions;										
≥100	user entries										
2	EcpCust	Displayed name of the menu entry.									
3	#	Always #									
4	#	Always #									
5	ECPMAIN_Help100	Internal Pro/E name of the quick help; it must start with ECP and must have the same number at the end as the label.									
6	help text: "calling ...\ecp353\tcl\cust\EcpCust.tcl"	Quick help (80 characters only, otherwise the menu item is not displayed).									
7	#	always #									
8	#	always #									

Block Structure of a Submenu

Note Submenus are only displayed if they contain at least one entry.

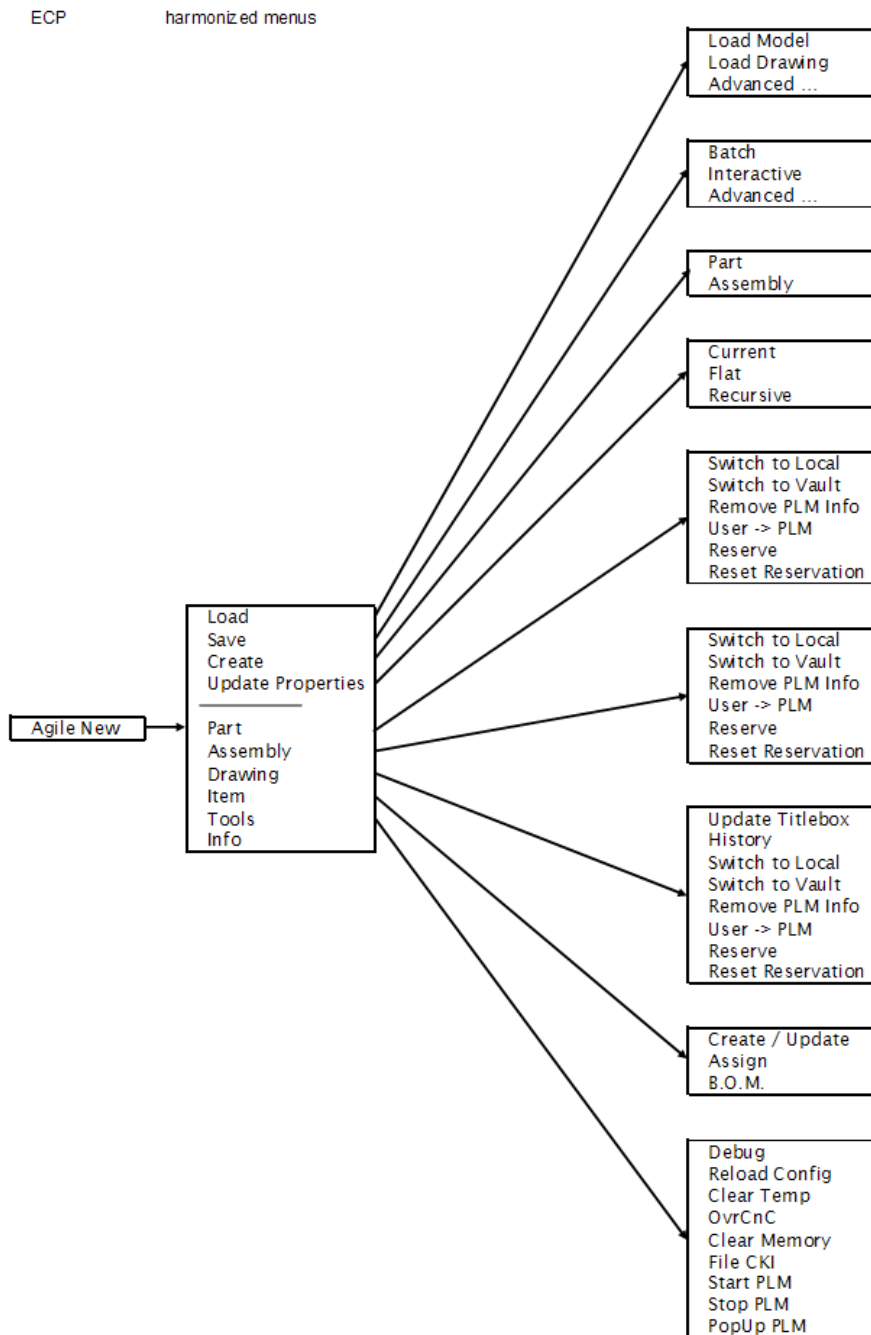
Line	Content	Description	Values	
1	ECPCUSTMAIN	Keyword for submenu ECPSUBMENU<number> When starting in the 2nd submenu level, the numbers are incremented (i.e. submenu 50 can generate submenus 501, 502, etc.)	Numbers of the 1st level (main menu):	
			up to 49	internal submenus
			50-99	user entries
2	Cust Menu	Displayed internal Pro/E menu name of the submenu entry		
3	#	Always #		
4	#	Always #		
5	ECPCUSTMAIN_Help	Keyword for resource file submenu ECPSUBMENU<number>_Resource		
6	ECP Cust Submenu	Name of the menu text file (with extension .txt)		
7	#	Always #		
8	#	Always #		

Harmonization of the Menu Structure

The integration's menu tree has been reengineered in order to align as well the menu structures as the menu names itself.

Redefining the setting "EcpMenuMainRes" with the content "ecpmain.txt" in the file EcpCustomer.ini permits to activate the structure as before.

Name	Default value	Values	Notes
EcpMenuMainRes	agilemain.txt	agilemain.txt ecpmain.txt	Activate the new harmonized menu structure. Activate the menu structure as before.



Parameter Transfer

This chapter describes the data transfer from Pro/Engineer to Agile e6 and vice versa.

Transfer from Pro/Engineer to Agile e6

Pro/Engineer parameters are transferred automatically via the software interface when the current object is saved. Based on the current object, parameters can be passed on to document master record fields.

There are five parameter ini files:

- 3d_model.par 3D model document type
- drawing.par Drawing document type
- item.par Item type
- 3d_model.min These files contain optional save information used in new functions for the 3D model document type.
- drawing.min These files contain optional save information used in new functions for the drawing document type.

These files are located in the directory defined by the variable pro_par_ini which is set in the file ECP_START.BAT. Its default value is set to %ECP_ROOT%\ini\par\.

Parameter ini files are used to define the data mapping in detail. They are not essential for a correct working of the software interface. Always, when such parameter ini files are used, all PLM system parameters with mode "A" must be specified

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

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

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.
------	---

If there is no previous record available for the current object in Agile e6, the default entries will be used as parameter values. This method ensures that even unknown objects can be saved in Agile e6 in a batch mode operation (automatic background processing). The transfer of Pro/Engineer parameter values to item records of the current object is analog. Pro/Engineer parameters are assigned to Agile e6 fields through parameter files.

When executing the "Save EDB" function, the user must choose one of two options: "Max Par" or "Min Par".

- "Max Par": The <name>.par files are used. (default setting)

- "Min Par": The <name>.min files are used for optional data.

Note The parameter transfer function can be disabled by renaming the parameter ini files.

Description of Parameter Entries

The parameter files have a unique and logical structure:

Example for 3D_model.par

Header	\$		-----						
	\$		Definition of created and updated data for document\$ 18.05.96 mie						
	\$		Mode A = Always (only system parameters); I = If exist; N = Never						
	\$		Type S = Standard; A = Attribute; P = Parameter; D = Dimension						
	\$		-----						
Table entries	\$		type	ident	mode	del	table	field	format
	\$		-----						
			[create section]						
			S	obj_typ	A	N	T_DOC_DAT	CAX_TYPE	"%s"
			S	loc_flg	A	N	T_DOC_DAT	CAX_LOCAL	"y"
			S	pe_release	A	N	T_DOC_DAT	CAX_CRE_SYSTEM	"Pro/E %s %s"
			P	Name	I	N	T_DOC_DAT	DOC_NAME	"%s"
			S	obj_nam	I	N	T_DOC_DAT	DOCUMENT_ID	"%s"
			[update section]						
			S	pe_release	A	N	T_DOC_DAT	CAX_CRE_SYSTEM	"Pro/E %s %s"
		[end]							

A **header** contains information such as the file description and the 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]. There is no limit with regard to the number of entries per 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.

[Create Section] and [Update Section]

Column	Value	Description	
type:		Defines the parameter type	
	A	Attribute	Only available in combination with Pro/DEVELOP or Pro/TOOLKIT Syntax: . 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 leading dot: <. Field> - field of the document master table Format: %s
	D	Dimension	Pro/Engineer dimensioning - reference by means of symbolic names Format: %f
	P	&Parameter	Pro/Engineer parameter - referenced via parameter names. If the "del" column value is set to "Y", the Pro/Engineer parameter is deleted after it has been retrieved. Format: %s Pro/Engineer YES/NO parameters are also supported. The return value is either "y" or "n" (for use with Agile e6 logic type fields).
	S	Standard parameter	Pre-defined parameter Format: various

Column	Value	Description	
	E	Environment variable	Retrieves an environment variable Format: %s
	Y	Symbol definition	Checks for occurrence of a symbol - Format: %s If found in a drawing sheet, A "y" value for the corresponding symbol is transferred to the logic Agile e6 field. If no symbol variant is found, an "n" value is returned and written to the field.
	G	Agile e6 data	Enables the transfer of Agile e6 document data from GENERIC to a variant (e.g. "variant inherits label of generic object."). Format: %s
ident:		Defines the parameter name	
mode:		Defines the execution mode	
	A	Always	Always transfer parameter (for system parameters only)
	I	If exists	Transfer parameter if exists
	N	Never	Never transfer parameter
del:		Identifier for deletion operations (i.e. parameter can be deleted) The "Delete" column is supported only for type P parameters. It identifies delete activities	
	N	NO	
	Y	YES	The parameter is deleted in Pro/Engineer once it has been transferred to Agile e6. The corresponding Agile e6 values are added to the "table" and "field" columns, i.e. the information entered for the Pro/Engineer name is the label field content. In case of a part you can view this information in the corresponding mask.

Column	Value	Description	
table:		Defines the Agile e6 table name	
field:		Defines the Agile e6 table field	
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.	
	Text	cre_sys	Currently creating system (system parameter)
	%f	density(*)	Density of current object
	%d	drw_sht	Total number of sheets of the Pro/Engineer drawing
	%s	drw_size	Drawing size of the current object
	%s	drw1_size	Like drw_size but limited to data read from Sheet 1.
	%s	frm_nam	Name of Pro/Engineer frame
	%s	frm1_nam	Like frm_nam but limited to data read from Sheet 1.
	%s	sht1_set	Switches to Sheet 1 of current drawing. No data is transferred to Agile e6. All subsequent parameters are retrieved from the drawing sheet.
	Text	frm_tab	Frame profile identifier (system parameter)
	y/n	loc_flg	Local flag default setting (system parameter)
	%f	mass(*)	Mass of current object
	%s	obj_nam	Pro/Engineer name of current object
	%s	obj_typ	Pro/Engineer type of current object (system parameter)
	%s%s.bmp	bmp_nam	Name of the bitmap file that Pro/Engineer creates for an object (PRT, ASM)

Column	Value	Description	
			Comment: This parameter must not be used outside of the "Update section".
	%s	pe_version	Pro/Engineer version
	%s	pe_revision	Pro/Engineer revision
	%s%s	pe_release	Pro/Engineer version and revision
	%d	pro_ver	Pro/Engineer version of current object
	%4.2f	scale	Drawing scale of the current drawing (Sheet 1) The drawing scale is set automatically according to the drawing size, i.e. "1:..." or "...: 1".
	%f	surface_area(*)	Surface of current object
	%s	unit_l	Dimension measurement unit (length) for current object
	%s	unit_m	Mass measurement unit for current object
	%f	volume(*)	Volume of the current object
	%f	x_max(*)	Maximum dimension of the current object in direction of the x-axis
	%f	y_max(*)	Maximum dimension of the current object in direction of the y axis
	%f	z_max(*)	Maximum dimension of the current object in direction of the z axis
			(*) These parameters are ignored in drawings

The following list contains examples of parameter definitions:

Value	Description
-------	-------------

"y"	Static text definition - in this case for the default setting "YES"
"%-10.2f"	Left-justified, ten digit numerical value with two places after the decimal point.
"%s"	Undefined character string (string)
"%2s"	Two-place character string
"%2.2s"	Two place character string, padded with a white space if required.
"%-4.0f"	Left-justified four-place numerical value without digits after the decimal point.
"%8D"	Eight-place date, (e.g. in "ecp.ini")

All parameter values start with the % character and have the syntax:

`%[<flag>][<field width>].[<precision>]<conversion sign>`

The flag is usually represented by a "-" character, i.e. the results of 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 conversion sign 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.

Example for Adding Customer Specific Parameters

The system administrator can configure the parameter files. This enables system operators to implement various company-specific configurations.

Here is an example dealing with dimensioning.

In order to select an appropriate packaging for a specific object, the maximum dimensions in a Cartesian coordinate grid (X, Y, and Z for length, width, and depth) must be known.

To this end, the coordinates must be added to fields in the corresponding table.

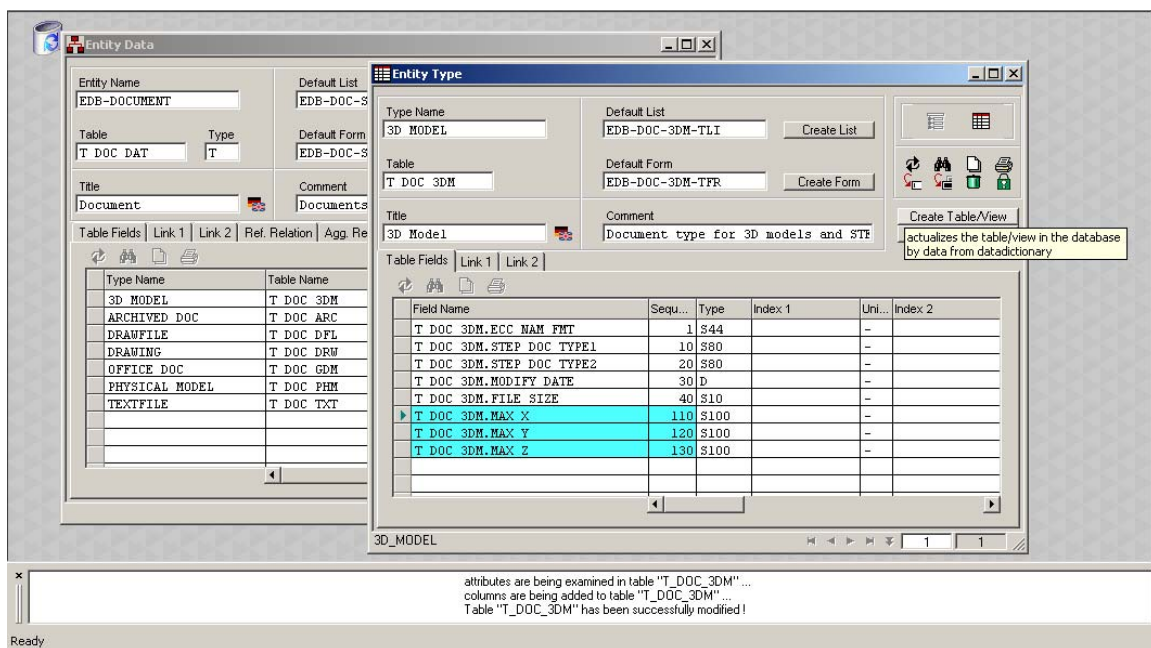
The following steps have to be carried out:

1. Select in Agile e6 the System Tables menu option in order to find the T_DOC_DAT table.
2. Select this table and open the Table-Field list via the field allocation.
3. Add the three fields

T_DOC_DAT.MAX_X
T_DOC_DAT.MAX_Y
T_DOC_DAT.MAX_Z

The default type is S100. Any enumeration may be used. However, in order to ensure good readability, increments of ten are recommended.

4. Modify the table with the Change Table command.



5. Use the System Masks menu option to assign the necessary entries to the corresponding masks.
6. Find the two masks EDB-DOC-3DM-TFR and EDB-DOC-3DM-TLI.
7. Select the Field Allocation option in the selected mask.
8. The Mask Field list is displayed.
9. To add a field, open the Table Fields list and use the drag & drop functions to move the corresponding fields to the other list.
10. Any enumeration may be used. However, in order to ensure good readability, increments of ten are recommended.
11. Go to the EDB-DOC-3DM-TFR form and specify the line and the column where the data is to be displayed in the form.
12. If you are uncertain about the exact location of the field, it may be best to use a random value first, and then position the fields in the mask generator. (Line 10 and Column 10 – 30)

Because no color definition has yet been defined for these fields, the new fields are displayed with a white background and therefore they can be easily identified.

13. Move the individual X, Y, and Z dimension fields to the desired locations in the mask.

Note You can also change the mask title.

14. Once you have finished the mask design, save it with the **Save > Database** menu option.
15. A confirmation request is displayed and you need to confirm the new enumeration of the field positions. This is not required and can be skipped with the **No** option.
16. Adjust the column in list view of EDB-DOC-3DM-TLI with the mask generator.
17. It is customary to define a unique name for each of the table column headings.
18. The displayed width of the column can also be adjusted. Do not define the virtual width too small. The default value of 100 is recommended.
19. Once the configuration is done, Agile e6 can be closed.

20. To initialize the fields that you have created in Agile e6 with values from the Pro/Engineer session, you need to modify the corresponding parameter file (in our example this would be 3D_MODEL.PAR.) using any text editor.
21. The aim is to transfer the dimension parameters whenever the object is updated (e.g. after a modification).
22. Adjust the [update section] accordingly.
23. If not already defined, add the corresponding lines. If the file already contains the required parameters, make them active (change from "N" = Never to "I" = If exist).

See also the example below:

[update section]

S	cre_sys	A	N	T_DOC_DAT	CAX_CRE_SYSTEM	"Pro/E %s %s"
S	x_max	I	N	T_DOC_DAT	MAX_X	"%-10.2f"
S	y_max	I	N	T_DOC_DAT	MAX_Y	"%-10.2f"
S	z_max	I	N	T_DOC_DAT	MAX_Z	"%-10.2f"

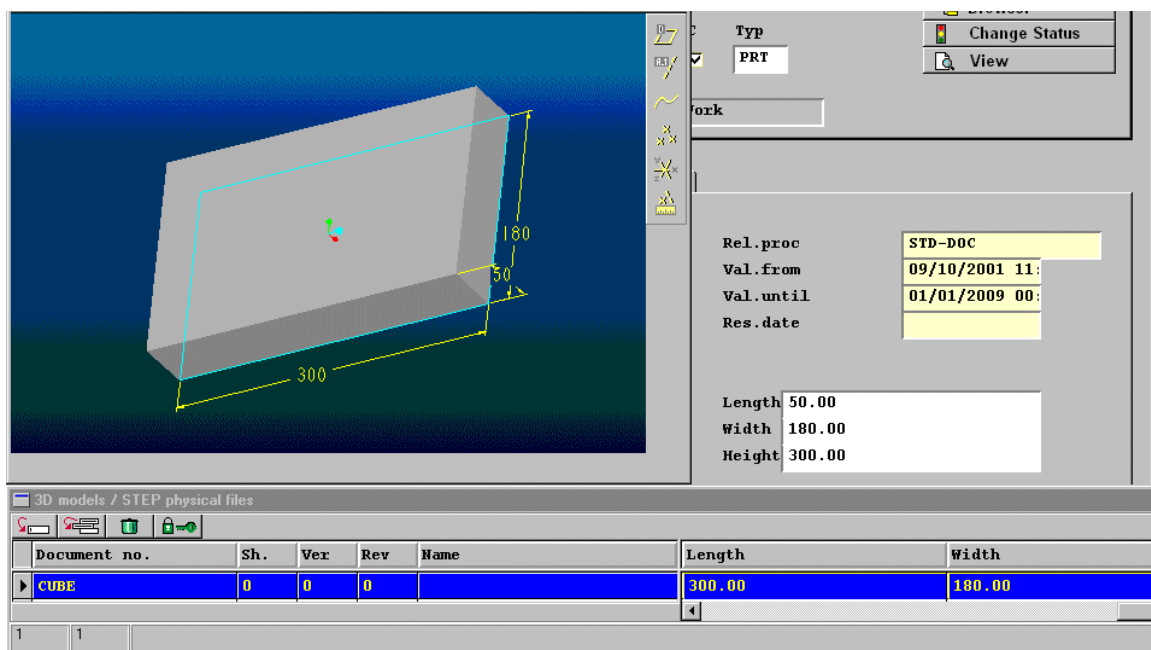
Note Enter valid parameters only. (See table [Create Section] and [Update Section] under entry forms)

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

24. Save the modified parameter file.

The changes will be available once the program has been re-launched.

The outer dimensions are now displayed for the Pro/Engineer object transferred to Agile e6. The values are automatically adjusted if the object is modified or saved in the vault of Agile e6.

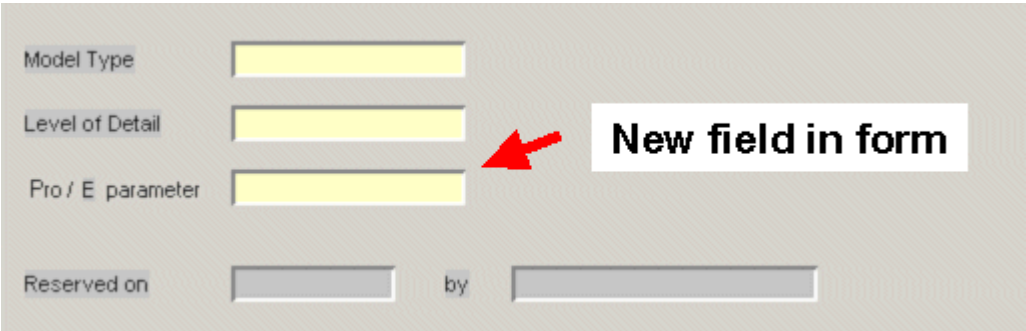


Example of a Pro/Engineer Parameter Transfer

Note For this example it is assumed that the new field name in Agile e6 is CAX_TEST

1. In Agile e6, add a field to the table T_DOC_DAT.
2. Add the field in EDB-DOC-3DM-TLI and EDB-DOC-3DM-TFR.
3. Position the field within the mask.

The headline for this new field in the 3D Model form could be “Pro/E parameter”.



The image shows a screenshot of a 3D Model form. It has four main input fields: 'Model Type', 'Level of Detail', 'Pro / E parameter', and 'Reserved on'. The 'Pro / E parameter' field is highlighted with a yellow background. A red arrow points to this field, and a white text box with the text 'New field in form' is positioned next to the arrow. The 'Reserved on' field is split into two parts: 'Reserved on' and 'by'.

4. To display this Pro/Engineer values in the new Agile e6 mask field, adjust the corresponding parameter file (3D_MODEL.PAR).

Assuming the parameter is to be transferred only when the object is created. If the Pro/Engineer file will be deleted afterwards, this value will be deleted in Agile e6 too. For this purpose, add **Y** in the **del** column (see example below).

\$-----						
\$type	ident	mode	del	table	field	format
\$-----						
[create section]						
S	obj_typ	A	N	T_DOC_DAT	CAX_TYPE	"%s"
S	loc_flg	A	N	T_DOC_DAT	CAX_LOCAL	"y"
S	cre_sys	A	N	T_DOC_DAT	CRE_SYSTEM	"Pro/E 17"
P	name	I	N	T_DOC_DAT	DOC_NAME	"%s"
S	obj_nam	I	N	T_DOC_DAT	DOCUMENT_ID	"%s"

P	TEST	I	Y	T_DOC_DAT	CAX_TEST	"%S"
---	------	---	----------	-----------	----------	------

Note Only I and N are valid mode values, since this is not a standard parameter.

Note Parameters that are to be transferred should be defined only once in the "create section" and set to Y in the del column. Parameters that are to be transferred repeatedly should be defined in the "update section" and set to Y in the del column.

- Define a parameter named TEST with a text entry in the Pro/Engineer session (or as defined in ident).

The value is transferred to Agile e6 as part of the check-in process as Current value and then it is displayed in the corresponding mask of the document.

Support of TCL Inside ECP

TCL enables the transfer of data between Agile e6 and the CAD system. All required libraries must be available on the operating system side. These libraries are located in the directory ...\\ecp35\\tcllib.

The directory ...\\ecp35\\tcl contains mapping files and TCL scripts related to the particular menu entries.

The additional capabilities of the TCL extension are used to extend or replace older LogiView procedures.

There are several ways to access TCL extension functions:

- Direct call via menu commands (e.g. Parameter function)
- Predefined menu options for custom extensions
- Execution as preaction or postaction of a standard function

By default, TCL is used for the following files and file types:

Mapping files for general software interface functions

EcpParameter.ini	"Parameter" function	mapping files
EcpAsmComponent.tcl	"Assemble EDB" function	TCL script
EcpExit.tcl	Software interface exit	TCL script
EcpInit.tcl	Software interface initialization	TCL script
EcpParameter.tcl	Pro/Engineer parameter writing	TCL script
EcpSetParameter.tcl	"Parameter" function	TCL script
EcpParameterCust.tcl	Customized Pro/Engineer parameter writing	TCL script

The configuration is implemented through (*.ini) mapping files, whose basic structure looks as follows:

[section name]
attributes = Value
...

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

The files are located in the directory specified in the variable "CaxConfigDir".

Transfer from Agile e6 to Pro/Engineer

There are three options to update the parameter of the current Pro/Engineer object (PLM > Object > Parameter).

- **Current** — Only the parameters of the current object will be updated.
- **Flat** — Only a single-level (current object and immediate childs) parameter will be updated.
- **Recursive** — The complete structure is processed.

If the current object is an assembly and the final option "*Recursive*" is used then the complete structure is processed.

Note If the value of a PLM parameter has been modified, the content will be updated in Pro/Engineer. Parameters that do not exist will be created automatically.

Note Pro/Engineer objects are considered to be modified once the parameters have been created or updated.

Note For parameters of instances with differing parameter values, the corresponding parameter will be added to the definition of the family table.

The file EcpParameter.ini contains all configuration data. Each parameter configuration is related to an object type.

[Initialize]	Additionally, parameters of related items can be retrieved if the variable "EcpSearchItem" is set to 1.
[EcpGetDrawingInfo]	The parameters that are read from the PLM document, for objects of DRW type, are defined in the "EcpGetDrawingInfo" section.
[EcpGetModelInfo]	The parameters that are read from the PLM document, for objects of ASM/PRT/LAY/MFG type, are defined in the [EcpGetModelInfo] section.
[EcpGetItemInfo]	The configuration settings for these parameters are defined in the "EcpGetItemInfo" section.

Note The source mask for the read operation must contain the assigned fields; however it is possible to set the visibility to "invisible".

[EcpGetDrawingInfo] Section Parameters

TYP	Entity_type = DRAWING	Document type for drawings (DRW)
MAS	Name_of_mask = EDB-DOC-DRW-TFR	Read operation source mask (form)
RET	RET: EDB_DOC_NAME = T_DOC_DAT.DOC_NAME RET: EDB_DOC_ID = T_DOC_DAT.DOCUMENT_ID RET: EDB_DOC_STATUS = T_CHK_STA.REL_STA	Parameter - field assignment Syntax: RET: <Pro/Engineer parameter name> = <table.field_name>

[EcpGetModelInfo] Section Parameter

TYP:	Entity_type = 3D_MODEL	Document type for models (ASM/PRT/LAY/MFG)
MAS:	Name_of_mask = EDB-DOC-3DM-TFR	Read operation source mask (form)
RET	RET: EDB_DOC_NAME = T_DOC_DAT.DOC_NAME RET: EDB_DOC_ID = T_DOC_DAT.DOCUMENT_ID RET : EDB_DOC_STATUS = T_CHK_STA.REL_STA	Parameter - field assignment Syntax: RET: <Pro/Engineer parameter name> =< table.field_name>

[EcpGetItemInfo] Section Parameter

MAS:	Name_of_mask = EDB-ART-SFR	Read operation source mask (form)
RET:	RET: EDB_ITEM_NAME = T_MASTER_DAT.PART_NAME RET: EDB_ITEM_ID = T_MASTER_DAT.PART_ID	Parameter - field assignment Syntax: RET: <Pro/Engineer parameter name> =

	RET: EDB_WERKSTOFF = BVB_WERKSTOFF.WERKSTOFFBEZ	<table.field_name>
--	--	--------------------

Transfer of PLM Data into Frames (Title Block) of Drawings

The software interface uses drawing tables. All data displayed in these tables are automatically retrieved from Agile e6 when the "Load Frame" function of Pro/Engineer is accessed.

The script ECP_START.BAT contains a reference to a files with a list of individual configuration file references. The default name of this file is ecp_fra.ini and it defines optional frame configuration files.

Note Each *.ini file corresponds with a table definition in the Pro/Engineer frame.

```
...
set cax_fra_ini=%ECP_ROOT%\ini\ecp_fra.ini
...
```

In the "ecp_fra.ini" each line references a configuration file and the corresponding storage location. Usually, these files will be located in the directory %ECP_ROOT%\ini\fra\.

1	.../ini/fra/standard.ini	"Standard"	"Standard ECP V 2.3"	EP
2	.../ini/fra/ecp_item.ini	"Item"	"Item data table"	EP
3	.../ini/fra/ecp_history.ini	"History"	"History log entries"	EP

Note These entries refer to example frames with a size of A4 that are delivered with the software interface.

- Line 1
Defines the standard title block. The corresponding configuration file is standard.ini.
- Line 2
Defines a table containing item data. The corresponding configuration file is ecp_item.ini
- Line 3
Defines a separate table containing history log entries that are additional to the entries which are already listed in the standard frame. The configuration file is ecp_history.ini

The individual lines have the following values:

Example	Description
1	The item number (consecutive numbers) of the particular entry is located in the first column.
/ini/fra/standard.ini	The path and name of the corresponding frame configuration file (path statements are relative to ECP_ROOT).

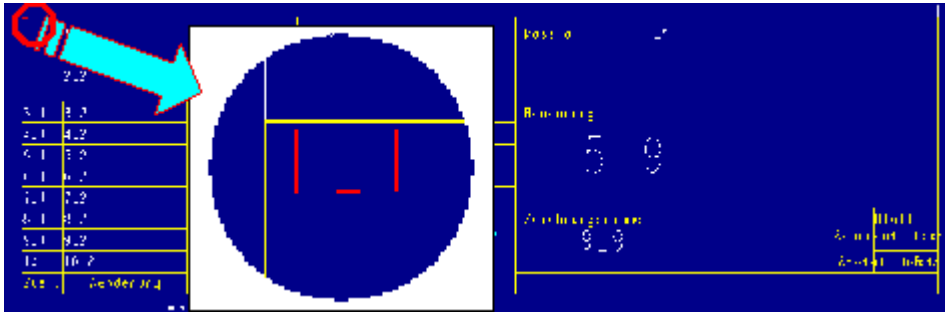
"Standard"	The menu option and notification text in quotes for the "Frame Change" function (is currently not supported).
"Standard ECP V 2.3"	For compatibility reasons
EP	The "EP" value defines the "CAX_SUBTYPE" and is used in Agile e6 filter functions. This variable is defined in the drawing parameter files, which are located in the directory path defined in the parameter "pro_par_ini". The parameter is defined in the file ecp_ini.*.

Configurations are assigned to tables via the table ident (table_ident). This is a character string of arbitrary length that must be located in the first column of line 1 of the Pro/Engineer table. If a table with a matching ident is found, then the Agile e6 data are written to this table.

It is possible to write data into several tables of the same type (e.g. multiple sheets).

The following table frame example illustrates the table write operation:

The configuration files are processed as described above. If a table with a matching table ident is found in Pro/Engineer then this data is written into this table.
In the example frame, the ident character string ("1_1") is located in the upper left corner of the table.



Here the character string "1_1" defines the table ident that is also specified in the standard.ini.

```
...  
# table_ident = 1_1  
...
```

The assignment of the entries of the configuration file standard.ini to the table fields take place using the Line_Column_ID entries in the corresponding cells.

All Pro/Engineer text format functions remain easy to apply.

However, the history field entries are an exception here. Relative column references are used for all history fields with the start field (start_field_of_history) as the origin.

During the write operation, the text format information is inherited from the cell text or - in case of an empty cell - from the global style definition.

The entries in the delivered standard configuration file standard.ini refer to the example frames delivered with the software interface. The Agile e6 information for a model with referenced items

and a drawing with the corresponding document history are positioned in the frame according to the standard configuration file.

Structure of Standard.ini

The default configuration file standard.ini is divided into the following sections.

Section	Description
[Global Section]	This section defines several commonly used parameter settings.
[Part Section]	This section defines part assignments in Agile e6. Do NOT make any changes to these entries!
[Document Section]	This section defines drawing information assignments for Agile e6 as well as for Pro/Engineer drawing heads. Document Section entries define the master record data types to be transferred from Agile e6 to the drawing
[History Section]	This section defines how history log of Agile e6 is displayed in the Pro/Engineer drawing title block.
[Special Section]	The Special Section defines special conditions for the transfer of data from the document history log to separate fields (e.g. Operator field).
[Total Section]	This section is not used in the software interface ECP
[Equivalence List]	This list defines the assignment of material data and model data for Agile e6 as well as for Pro/Engineer title boxes. The Equivalence list has two functions: - It controls the predefinition of fields depending on item, model, and drawing EDB objects. - It defines how model and item specific EDB attributes are displayed in the drawing frame.

[Global Section]

Section	Description
part_type	Document type of the part file
drawing_type	Document type of the drawing
start_field_of_history	Start field of history log entry in the title block
cadim_mask	Read operation source mask
update_frame	Frame handling method for drawing load operations 1 → Do not load any frames

	2 → Load frame if drawing contains one 3 → Always load frame
logiview_action	Post-LogiView-action (compound/procedure) No longer supported. Do NOT modify.
history_records	Number of displayed history records
history_prefix	Prefix of history log entity names
history_data	Number of history log fields in the history section
history_offset	Do NOT modify
history_function_1	Do NOT modify
history_function_2	Do NOT modify
read_widget_mode	Widget mode for read operations form: form mode list : list mode
cad_flag_only	Always set to y
file_manager	Always 1
permanent_load_list	1 : Use permanent load list 0 : Always create new load list
checkin_area	Not used
table_ident	Drawing table ID

[Document Section]

table_name.field_name	T_DOC_DAT.DOC_NAME	Describes the PLM table and field for the content of the document name field in the title block.
Line Number_Column Number	5_9	Refer to the title block (drawing table) defined in the frame. A zero (0) marks those fields that contain information that must not be transferred from the document master record.
format	"%-12.12s"	With the 'format' command, you can influence the appearance

		of text in the frame. This is done using standard "C" syntax.
--	--	---

Note The first part of the Document Section should not be modified. Additional data can be added in the second part called "Optional configuration parameters".

Note The last entry T_DOC_DRW.CRE_USER must not be deleted!

[History Section]

'table_name.field_name'	T_DOC_HIS.MODIFY_NAME	The location of the history start field is defined in the parameter start_field_of_history in the Global Section.
'Column Number'	4	The column numbers are counted in ascending order with the start field as the origin.
'format'	"%-4.4s"	With the 'format' command, you can influence the appearance of text in the frame. This is done using standard "C" syntax.

Note The history start field is used as the origin.

The following parameters **must not be modified** in order to ensure the proper operation of the function

- T_DOC_HIS.FUNCTION
- T_DOC_HIS.CHANGE_STATUS
- T_DOC_HIS.HIST_ID

[Special Section]

'table_name.field_name'	T_DOC_HIS.FUNCTION	This value defines the field name in connection with a specific value.
'value'	"INSERT"	Describes the possible access methods.
'table_name.field_name'	T_DOC_HIS.MODIFY_NAME	Describes the PLM table and field for the content of the name field of the history in the title block
'line_column'	4_7	Describes line and

		column where the info is to be placed in the title block.
flag	0	The "flag" specifies the occurrence of data: 1 = last occurrence 0 = first occurrence
'format'	"%-8.8s"	With the 'format' command, you can influence the appearance of text in the frame. This is done using standard "C" syntax.

[Equivalence List]

Describes the field mapping of associated item data to the title block.

Note	The first line is a comment line! It contains the names of the default tables for the "table_name" entry.
------	---

Note	Ensure that all the data you want to transfer are available in the referenced mask fields of the EDB mask!
------	--

Note	The property "Visibility" of such mask fields can be either "visible" or "invisible" in order to maintain a good appearance of the mask.
------	--

Using Custom Drawing Frames

The ECP 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.

Inside the drawing frame mode you can use so-called "Drawing Tables" to design the title blocks. In Pro/Engineer, drawing tables can be saved as separate files and later be reused for other format sizes (FILE_NAME.TBL). The table must always contain a table ID for proper recognition by the software interface. The cell in Line 1 / Column 1 of a standard table contains the string "1_1". The only purpose of this string is to identify the table and its size. Therefore, it can be very small.

First of all, you need to define a new standard format with a unique name in Pro/Engineer. Once the format size, position, and unit have been defined, the outline of the new sheet is displayed on the screen. Use the functions "Details zzzzz" and "Sketch zzzzz" to define the format geometry.

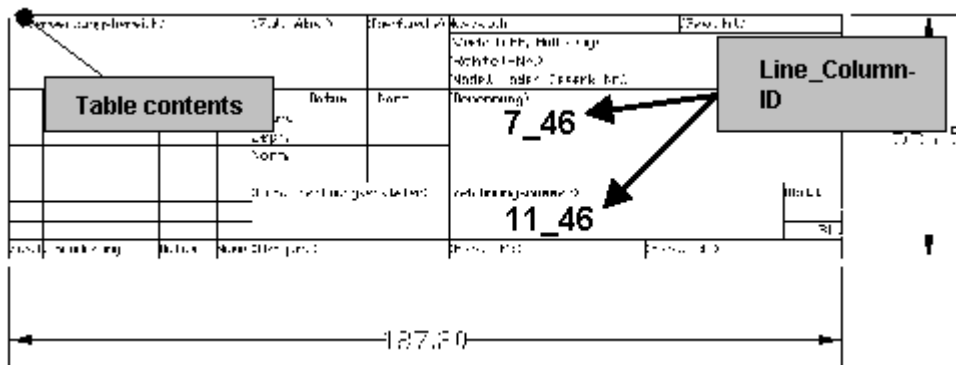
The next step is to create a drawing table. This table must fit into the previously defined frame and

should contain text information retrieved via the software interface.

You can also add parameter identifiers such as "&format" or "t_sheet".

Note The text format information (such as font types, font size, and color) has to be defined for all fields containing text that is retrieved via the software interface ECP.

Example: title box.



Using LogiView

LogiView is a proprietary programming extension inside Agile e6 that enables the implementation of project specific functionality in the software interface. In order to support the transfer of data from Agile e6 to Pro/Engineer, some LogiView procedures are already included by default. It is common to use several LogiView procedures.

Data transfer is performed using a transfer file. The path and name of this file (but not its extensions) are defined by the value of the environment variable `cax_lgv_dat` in the file `ECP_START.BAT`. The default entry is:

```
set cax_lgv_dat=%cax_temp%\lgv_dat .
```

The corresponding menu text is defined as the value of the environment variable `cax_lgv_men` in the file `ECP_START.BAT` file. The default entry is

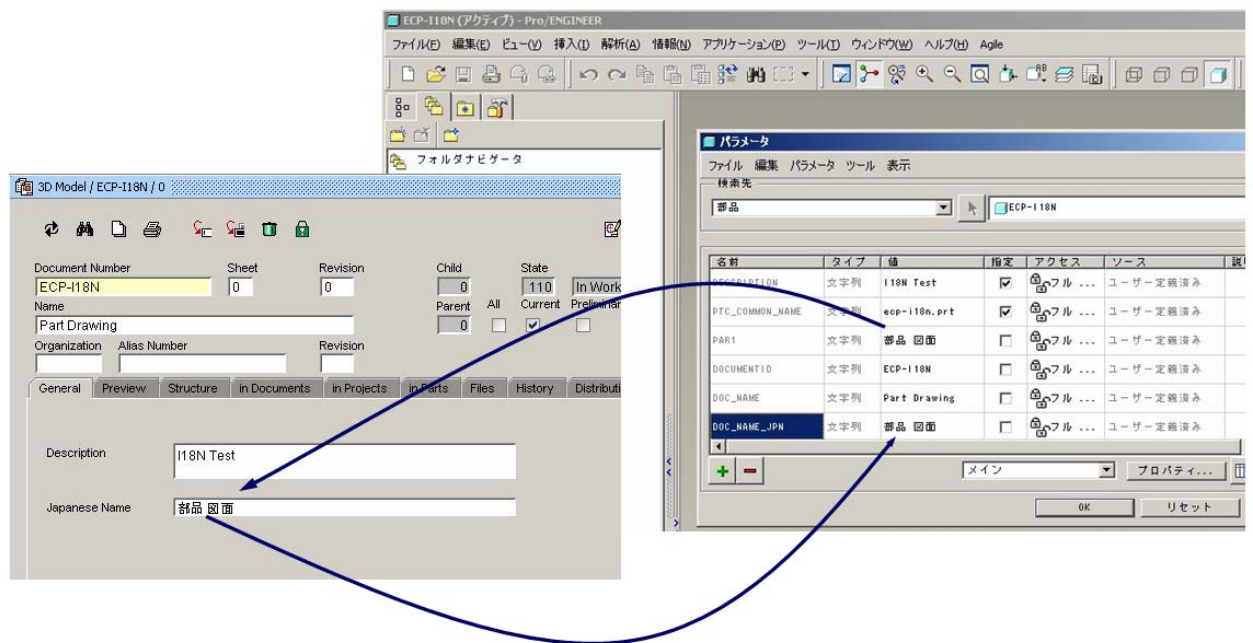
```
set cax_lgv_men=%ECP_ROOT%\ini\lgv.ini .
```

Chapter 6

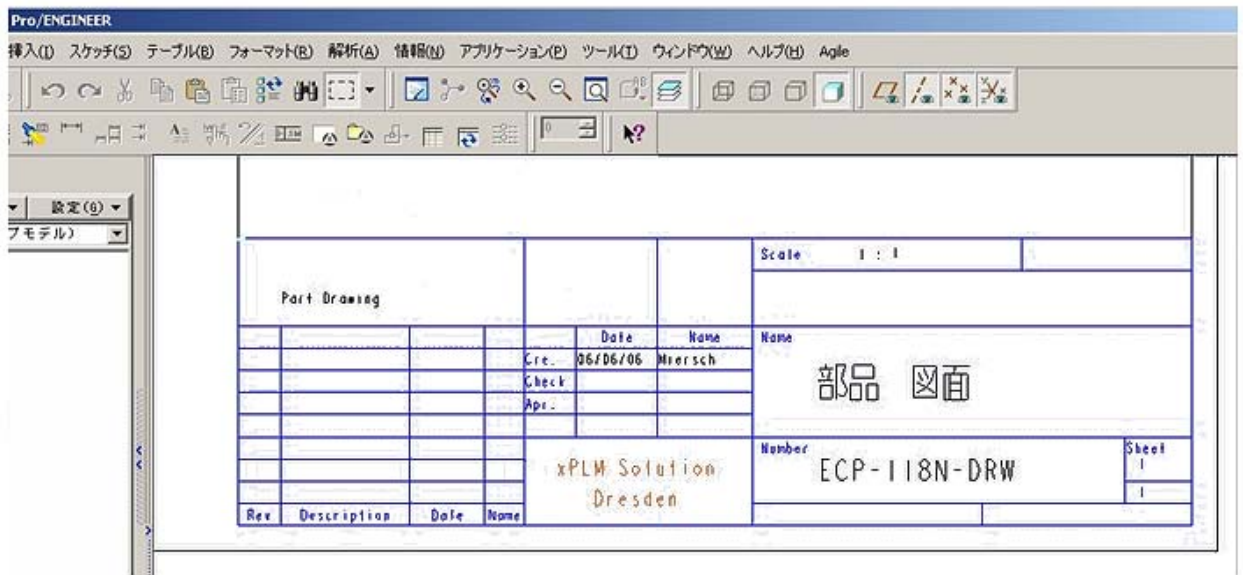
I18N and L10N

I18N Basic Support

Agile e6.0.2 enables the transfer of parameter content from Pro/ENGINEER to the PLM system and vice versa. This is the realization of the basic concept of internationalization (I18N) for the integration software ECP, because this feature can be used to support parameter content with international characters.



Enables the transfer of I18N parameter content into a drawing frame.



For I18N, two new settings are included in the file EcpCustomer.ini as comments. They must be activated if needed.

Name	Default Value	Value	Description
EcpI18N Support	1	0	Activates I18N basic support when using ECP
EcpProEncoding	A string to describe the encoding, which is used in ECP		

Additionally, in the mapping file EcpParameter.ini, those sections have to be included that are related to transfer the I18N parameter content.

L10N Software Design Features

It is possible to realize the design concept of localization (L10N) especially for software, which is used in Windows and Unix, according to customer's preliminaries.

The following possibilities enable the integration software.

- All text resources are located in language depended directories.
- Using the content of the environment variable ECP_LANG you can determine the actually used language.
- Usage of internal language invariant labels for the dialogs that are used in Pro/ENGINEER is possible starting with Wildfire 2.

Chapter 7

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 8**Appendix****Including Your Own Menu Entries / Menus**

In the following example, menu entry "Ecp Cust" is added to the main menu.

The required extensions to the text files for this example have already been prepared; they are located in directory ...ecp35\addons\Menu.

1. Add a new block to the end of ecpmain.txt:

```
...
ECPMAIN_Entry100
Ecp Cust
#
#
ECPMAIN_Help100
Call up Ecp Cust script
#
#
```

The integration automatically generates the internal menu entry "EcpMainMenuEntry100" from this after a restart.

2. Create a subdirectory "cust" in the TCL directory of the integration.
3. Save the script "EcpCust.tcl" to this directory.

The script "EcpCust.tcl" is assigned to the menu entry "EcpMainMenuEntry100" in EcpCustomer.ini using the "EcpMenu" section:

```
...
[ EcpMenu ]
EcpMainMenuEntry100      =  cust\EcpCust.tcl
...
```

If mapping does not occur, the integration is trying to call up a script with the name "EcpMainMenuEntry100.tcl" (assumed directly located in TCL directory) which cannot be found.

The entry:

EcpIcon ECPMAIN_Entry100 = <name>.gif

in EcpCustomer.ini ("Initialize" section) can be used to assign an icon to the menu item and the function can be made available within Pro/E Wildfire.

Including Your Own Submenus

In the following example, submenu "Cust Menu" is added to the main menu. The required extensions to the text files for this example have already been prepared; they are located in directory ...ecp35\addons\Menu. (file ecpmain.add).

... ECPSUBMENU100 EcpCustMenu # # ECPSUBMENU100_Resource ecpcust.txt # #	Add a new block to the end of ecpmain.txt:
ECPCUSTMAIN Cust Menu # # ECPCUSTMAIN_Help ECP Cust Submenu # # ECPCUST_Entry1 Ecp Cust # # ECPCUST_Help1 Sample entry for submenu # #	Create file ecpcust.txt in the text path of the integration with the following content (or copy it from ecp35\addons\Menu to the text path):
The integration automatically generates a new submenu with an entry of "EcpCustMenuEntry1" when the next restart happens.	
EcpCustMenuEntry1 = cust\EcpCust.tcl	Add this entry into file EcpCustomer.ini section [EcpMenu].
EcpCustMenuEntry1 = cust\EcpCust.tcl	Add this entry into file EcpCustomer.ini section [EcpMenu].
EcpIcon ECPCUST_Entry1 = <name>.gif	<p>Add this entry into file EcpCustomer.ini section [Initialize].</p> <p>An icon can be assigned to the menu item and the function can be made available within Pro/E Wildfire .</p>

Overview of ECP_Master.ini Settings

Name	Default Value	Values	Description
EcpAddComponentMenu	1	0 1	Activates menu extension for Pro/E ASM component menu to insert a component using ECP.
EcpAskSimpRep	1	0 1	When saving a module with a „Simplified representation“, a warning appears.
EcpBatchIniFile	EcpBatch.ini	Name	Name of the initialization file used in the plot module (ECP-PLT).
EcpBomMenuActionDefault	Batch	Batch Int	Controls the default value „Interactive“/“Batch“ in the "BOM" dialog box.
EcpCaxEciCount	0	Number	Number of ECI calls until reconnect. If this is not set or is set to 0, it is inactive.
EcpCheckCreSystem	commented out	Check string	The check string is used to check the generating system when retrieving objects. The value of the generating system field is checked for the number of letters defined in the check string. Permitted values in check string "Pro/E" are, for example, "Pro/E 2000i2" or "Pro/ENGINEER 2001 2002030"; not permitted is, for example, "PRO/E". Objects that are not permitted are skipped.
EcpCheckDrwStructureFlag	0	0 1	When drawings are retrieved, the structure flag (T_DOC_DAT.CAX_STROK) is selected. If the field content is "n", the drawing and the models are retrieved using "Load „Simplified representation" (1. check out files; 2. retrieve drawing).
EcpCheckObjInSession	0	0 1	Checks before retrieval if an object with the same name (with a different or missing c_id) has already been loaded into a Pro/E session. In this case the PLM info is not automatically assigned to this object, i.e. it remains a user object.

Name	Default Value	Values	Description
EcpCheckVerViewRetrieve	0	0 1	Check of the version view before retrieval. If the current version view in PLM is "Global", retrieval is cancelled and an error message is displayed.
EcpCheckVerViewSave	0	0 1	Check of the version view before saving. If the current version view in PLM is not "Development", saving is cancelled and an error message is displayed.
EcpCreateInstanceMsg	NONE	PLM message	A message is displayed in the PLM message window when an instance is created.
EcpCreateLastSaveBaseline	0	0 1	Activates/deactivates automatic creation of a LastSave baseline (document baseline).
EcpCreateMsg	NONE	PLM message	The message is issued in the PLM message window when an object is created.
EcpCreateSimpRepBaseline	0	0 1	Activates automatic creation of baselines for „Simplified representations“ (document baseline).
EcpDebugLevel	0	0 1 2	Sets the debug level of the TCL extension. The value is set automatically when ECP Debug is activated. 0 = Debug off; 1 = Debug on; 2 = Write supplemental debug info
EcpDeleteDrwRelation	0	0 1	Model-drawing relationship is deleted and newly created during every save procedure.
EcpDeleteExtRefRelation	0	0 1	Attempt is made to delete any external relationships that may exist during every update to a PRT.
EcpDeleteFmsFiles	Session	All Session None	This affects the deletion of FMS files in the Temp directory of the integration. None = Files are not deleted from tmp.

Name	Default Value	Values	Description
			Session = Only the files that are open in the current session are deleted. All = All files are deleted.
EcpDeleteLocalCopy	1	0 1	Deletion of local files when switching from local to FMS.
EcpDeleteSpecialSectionTable	1	0 1	Table entries in the drawing frame, which are filled using the "special section", are deleted before the drawing frame is updated.
EcpDocumentIni	EcpDocument.ini	Name	File name of the configuration file for documents (must be located in path <CaxConfigDir>).
EcpEciHost	localhost	Name	Host name to which the ECI connect is directed.
EcpFlatAsmInstances	1	0 1	Only single stage of nested family tables (assembly) is displayed.
EcpFlatPrtInstances	0	0 1	Only single stage of nested family tables (part) is shown from the integration.
EcpHelpPartFlat	0	0 1	The substructures of auxiliary parts are not included in the BOM. Such assemblies and all its substructures are skipped when generating the BOM.
EcpHelpPartIdent	KENNWORT	Name	Defines the parameter name for auxiliary parts.
EcpHelpPartValue	HT	Name	Defines the parameter content for auxiliary parts. Objects that are marked in this way are ignored when creating items or generating the BOM.
EcpI18NSupport	1	0 1	Activates I18N Basic Support when using ECP.
EcpIcon*		Name of a gif file	Assigns integration functions to icons and enables them as internal Pro/ENGINEER functions (Pro/ENGINEER Wildfire or later).

Name	Default Value	Values	Description
EcpInfoCustParameter	NONE	Name	If the name of a Pro/ENGINEER parameter is set, this name is shown in the last column in the Info function window.
EcpLastSaveBaselineMode	Full	Full None	Mode FULL forces the creation of LastSave baselines with only one entry.
EcpMenuMainRes	ecpmain.txt	Name	Name of text resource for the main menu of the integration (text file in txt directory).
EcpNoAsk	0	0 1	Deactivates all queries for batch plotting. Must be set to 1 in EcpBatch.ini.
EcpNoFrameUpdate	0	0 1	Deactivates automatic frame update when retrieving drawings.
EcpNoItemCreate	0	0 1	Prevents the creation of items using the connector.
EcpNoItemUpdate	0	0 1	Prevents the update of items using the integration.
EcpNotCheckConcurrentAccess	0	0 1	Deactivates checking for parallel access (concurrent check).
EcpNotCheckInstances	1	0 1	Length of instances is checked.
EcpNotOpenInstances	0	0 1	Instances are not opened in the Pro/ENGINEER session during check-in (works only if EcpFlatAsmInstances = 1 or EcpFlatPrtInstances = 1).
EcpNotReadUserName	0	0 1	Instead of the user name, the user ID is displayed in the ECP Info function window.
EcpNotRenameLayout	0	0 1	The layout is not renamed during the first check-in.
EcpNotRenameObject	0	0 1	Objects are not renamed.

Name	Default Value	Values	Description
EcpNotUseTotalHistory	0	0 1	Deactivates reading of the complete history to fill out the history block in the frames (if set to 1).
EcpNoWinPop	0	0 1	Deactivates window handling while batch plotting. Must be set to 1 in EcpBatch.ini.
EcpParameterIniFile	EcpParameter.ini	Name	Name of the ini file that is used for the parameter update.
EcpRenameFileOnDisk	0	0 1	When renaming checked in files, also rename files at the original (local) location.
EcpRenameInstance	1	0 1	Rename instances at the first check-in.
EcpReusePartValue	WT	Name	Defines the parameter content for standard parts. The objects that are marked in this way are ignored during normal saving. The parameter name is defined using EcpStandardPartIdent.
EcpSaveInstanceAccelerator	0	0 1	Support of instance accelerator files (xpr/xas).
EcpSaveInstanceBmp	0	0 1	BMP files have to be saved to create BMP files of instances. Activates the saving of instances in the integration.
EcpSaveMenuActionDefault	batch	Batch Int	Controls the default value „With further inquiry“/ „Without further inquiry“ in the "Save" dialog box.
EcpSaveRenameError	0	0 1	Created document is deleted, save procedure is cancelled (without any query) and the error list is displayed when the error occurs.
EcpSelectReturnCheck	commented out	off	It is not checked from which form the selection "Select to CAD" was released.
EcpSetTypeSelect	0	0	Do not use.
EcpSetWindowAct	commented out (RST +	POP	Affects the pop-up action of the PLM window while using the

Name	Default Value	Values	Description
	POP)	RST MAX	integration.
EcpSetWindowLow	commented out (ICO)	ICO LOW	Affects the action when PLM window is open in the back while using the integration.
EcpSimpRepBaselineMode	Current	Current All	<p>“Current”: creates only a baseline for the current „Simplified representation“ after saving if the active assembly is currently loaded as „Simplified representation“. No action occurs in the MASTER display.</p> <p>“All”: after saving in the MASTER display a baseline will be created for all „Simplified representations“. If the active assembly is currently loaded as „Simplified representation“, a baseline is created only for this current „Simplified representation“.</p>
EcpStandardPartIdent	Kennwort	Name	Defines the parameter name for reuse and standard parts.
EcpStandardPartValue	NT	Name	Defines the parameter name for norm parts. The objects that are marked in this way are ignored when normal saving.
EcpStartObjectName	standard	Name	Standard object name; defines the default name for start objects.
EcpSwitchDrwRelation	0	0 1	The drawing document is created in the “Where used” tab of the model document.
EcpSwitchInstanceRelation	0	0 1	The instance document is created in the “Where used” tab of the GENERIC document. The GENERIC document is located in the structure of the instance document.
EcpSwitchLayRelation	0	0 1	The PRT/ASM document is created in the “Where used” tab of the layout document. The layout document is located in the structure of the PRT/ASM

Name	Default Value	Values	Description
			document.
EcpTimer	0	0 1	Activate output of timer information.
EcpUI*Dialog		Name	Name of the dialog box resource used for a dialog box (without file extension).
EcpUpdateInstanceMsg	NONE	PLM message	The message is issued in the PLM message window when an instance is updated.
EcpUpdateMenuActionDefault	Batch	Batch Int	Controls the default value „Interactive“/“Batch“ in the function "PLM -> User".
EcpUpdateMsg	NONE	PLM message	The message is displayed in the PLM message window when an object is updated.
EcpUpdateNoSave	0	0 1	Controls the default value of option "Save" in function "User → PLM".
EcpUseBaseline	0	0 1	Activates the use of the baseline feature in ECP.
EcpUseBomComponent	commented out	0 1	An entry is made in the BOM when it is generated when a component is used several times.
EcpUseDocStrComponent	0	0 1	Components that are repeatedly integrated within the same level in the document structure are presented several times as well.
EcpUseFastRetrieve	0	0 1	Activate/deactivate the use of the fast retrieve function.
EcpUseFileBaseline	0	0 1	0 = Generates a pure document baseline when saving a baseline. 1 = Creates an additional file assignment when saving a baseline.
EcpUseHistoryTrn	1	0 1	"Memos" are not evaluated when reading the history for the concurrent check.
EcpUseNewConcurrentChe	0	0	The check for parallel access (concurrent check) is made using

Name	Default Value	Values	Description
ck		1	the version stamp and no longer using the history (works only if EcpUseVerstamp = 1).
EcpUseNewMenu	1	0 1	Activates/deactivates the use of the new menu structure.
EcpUseOldGetParameter	0	0 1	Old functions (Pro/DEVELOP) are used to access parameter contents.
EcpUseStandardPartSubType	0	0 1	Content of field T_DOC_DAT.CAX_SUBTYPE is also considered during the detection of standard parts.
EcpUseVerstamp	0	0 1	The Pro/ENGINEER version stamp is used to determine if an object has been modified during the session.
EcpWindowActivateAction	1	0 1 Short	A short info for "Change window" for the current object is displayed in the message window. 0 =off / 1 =on / SHORT = Only the information "PLM Object" or "No PLM Object" is displayed; no query is made in PLM.
EcpGetAddErrorInfo	0	0 1	Causes extended ECI error messages if activated.
EcpCreDefUseSpace	0	0 1	Enables to delete PLM-field content by transferring a space character if activated.

Additionally, a brief help description for each function is shown in the Pro/Engineer message window.

