

**Freedom/Plex
Integrating CA 2E to CA Plex ERD
User Guide**

Version 4.3

HawkBridge Pty Ltd
3 Highett Road
Hampton, VIC 3188
Australia

<http://www.HawkBridge.com.au>

Copyright © 2011 by HawkBridge Pty Ltd

All rights reserved, including the right of reproduction in whole or in part in any form



Table of Contents

Overview..... 3

Shareware 3

Prerequisites..... 4

 OS/400 V5R2M0 or above.....4

 CA 2E 8.0, 8.1 or 8.54

 CA Plex 5.5 or above4

Using the Freedom/Plex Integration Tool..... 5

 Create the Plex Import File5

 Transfer the CA Plex Import File to the PC5

 Import File into CA Plex.....7

Interface Issues and Design Standard..... 10

 Freedom/Plex Object Type Cross Reference10

 Freedom/Plex Export Layers.....10

 CA Plex Attribute and System Layer Object Names11

 CA Plex Object Names11

 CA 2E Physical Access Paths12

 CA 2E Span Access Paths.....12

 CA 2E *NONE Access Paths.....12

 CA 2E Includes Relations13

 CA 2E Entry Redirection for Denoted by, Includes, Known by, Qualified by, and Has Relations13

 CA 2E Relationship Checks and Overrides13

Command Reference..... 14

 Create Plex ERD Import File (HCRTPLXERD).....14

Overview

Freedom/Plex provides a convenient method of transferring data model information from CA 2E data models to CA Plex data models where applications can be generated for multiple target platforms.

CA Plex is the natural path for migration for many CA 2E users. Currently there are methods to port the database relations and functions as API's from CA 2E to CA Plex. This provides users with the capability of developing in a co-existence environment. Freedom/Plex provides a method of porting the application areas, files, fields, relations, conditions, access paths, arrays, and messages from a CA 2E data model into CA Plex.

Migrating to CA Plex provides a means of preserving your investment made in CA 2E. For some organisations this could equate to 100+ man years of business logic development and many millions of dollars in investment. Developers increase their productivity in CA Plex where inheritance and patterns provide a level of abstraction not possible in CA 2E. CA Plex also enables applications to be generated for a greater number of target platforms providing portability for business logic.

Shareware

This product is distributed as Shareware. You may install and use the product for 45 days on a trial basis. Should you wish to use the product after the 45 day trial you are required to purchase a license from HawkBridge Pty Ltd for your AS/400. The authorization will then enable you to install and use the product on a single AS/400 only.

Shareware products are provided without warranty either implied or expressed. It is your responsibility to ensure that the software is appropriate for your installation.

This shareware product has been restricted to only allow a attribute and system layer CA 2E objects to be imported into CA Plex. The fully licensed version has no restrictions imposed apart from running on a single nominated AS/400.

Prerequisites

The following prerequisites must be satisfied in order to use the tool:

- OS/400 V5R2M0 or above installed
- CA 2E 8.0, 8.1 or 8.5 Data Model
- CA Plex 5.5 or later

OS/400 V5R2M0 or above

The AS/400 objects in this release have been saved for V5R2M0. If you are on an earlier release, then contact us for a version that supports your version of OS/400.

Objects have been saved in a format that will enable Freedom/Plex to be restored to V6R1 and V7R1 of OS/400.

CA 2E 8.0, 8.1 or 8.5

The tool has been developed for Release 8.0, 8.1 and 8.5 of CA 2E. If you are on a later release, then contact us for the upgraded version of the tool. Prior releases may work, but may not due to internal file changes to the CA 2E data model.

CA Plex 5.5 or above

The tool has been developed for CA Plex 5.5 or later.

Earlier versions of CA Plex are not supported in this product.

Using the Freedom/Plex Integration Tool

To use the Freedom/Plex integration tool to import CA 2E data relationship and definition model objects into CA Plex follow these steps:

Create the Plex Import File

Make sure that FREEDOMPXE (or the library you installed Freedom/Plex in) is in your library list. The minimum library list to run the interface tool is:

- QTEMP
- < *MDLLIB >
- < *GENLIB >
- Y2SVVENG
- Y2SY
- Y1SYVENG
- Y1SY
- QGPL
- FREEDOMPXE

Where < *MDLLIB > and < *GENLIB > are your data model and generation libraries respectively.

Libraries Y2SYVENG and Y1SYVENG may have been merged into Y2SY and Y1SY respectively when installed. In which case, they should be omitted from the initial library list.

Prompt the command HCRTPLXERD and enter the following details then press Enter:

```

Create Plex ERD Import File (HCRTPLXERD)

Type choices, press Enter.

Model object name:
  Object owner--*generic . . . . > *ALLOBJ_____
  Object name--*generic . . . . _____
  Object type . . . . . _____ *ALL, *ACP, *APP, *ARR...
Model object surrogate . . . . *OBJNAM_____ Number, *OBJNAM
Model object list for copy . . . HPLXCPYLST Name, *SELECT, *S
  Model library name . . . . . *MDLLIB_____ Name, *MDLLIB
Edit copy list . . . . . *NO_____ *NO, *YES
Import file . . . . . HPLXIMPDTA Name
  Library . . . . . *GENLIB_____ Name, *GENLIB
Import member name . . . . . *FILE_____ Name, *FILE
Extract options . . . . . *ALL_____ *ALL, *ATRALL, *ATR...
  + for more values _____
Attribute group model name . . . *NONE_____ Character value, *NONE
System group model name . . . . *NONE_____ Character value, *NONE

Bottom
F3=Exit F4=Prompt F5=Refresh F12=Cancel F13=How to use this display
F24=More keys

```

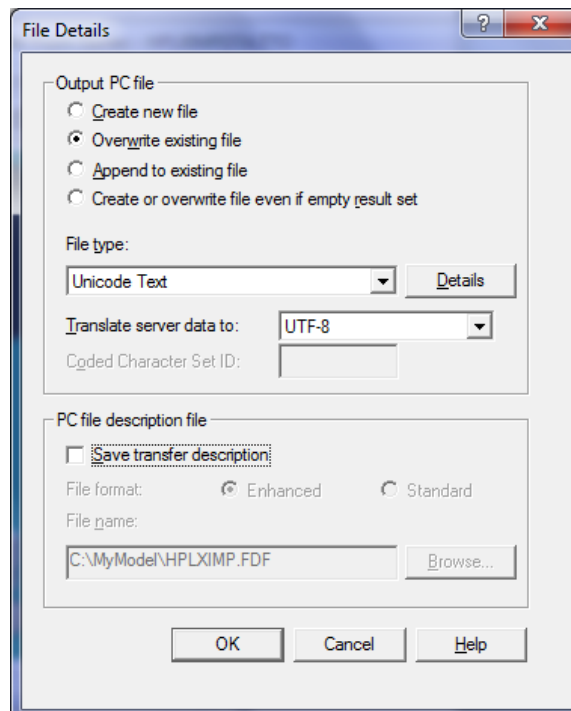
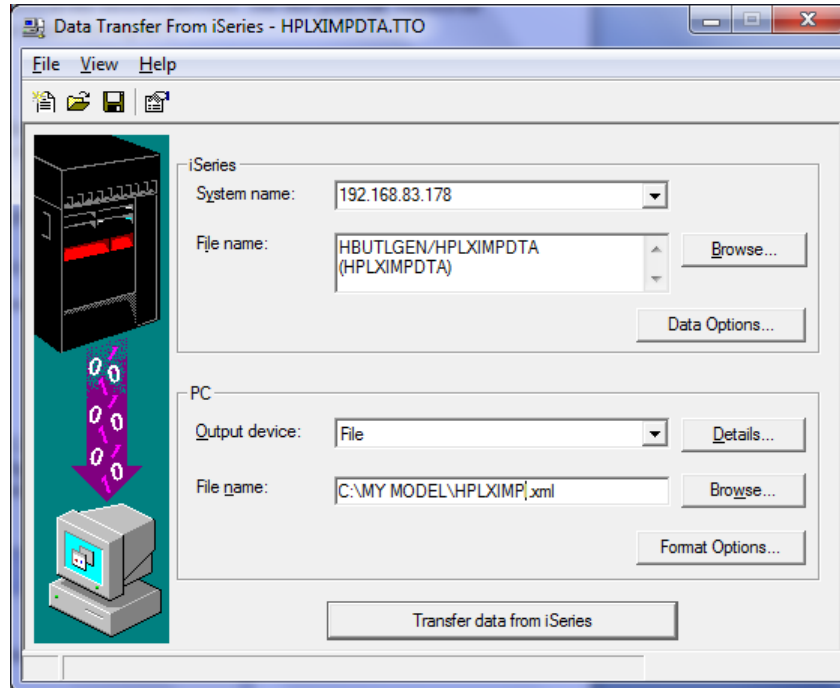
Note: This command is long running and should be submitted to batch for more efficient processing. Even when only a few objects are selected, the command will navigate all objects in the data model twice. The first pass will expand all objects for processing and the second pass will generate the details for export.

Transfer the CA Plex Import File to the PC

Start an AS/400 to PC file transfer session to transfer the file HPLXIMPDTA member HPLXIMPDTA in the generation library to a temporary PC directory, such as C:\TEMP. The transfer details are as follows:

- From file: <*GENLIB>/HPLXIMPDTA(HPLXIMPDTA)
- To File: C:\TEMP\HPLXIMP.XML
- To File Type: Unicode (UTF-8) or ASCII Text (Code Page 1208)

Note: <*GENLIB> should be replaced with the actual name of the generation library.



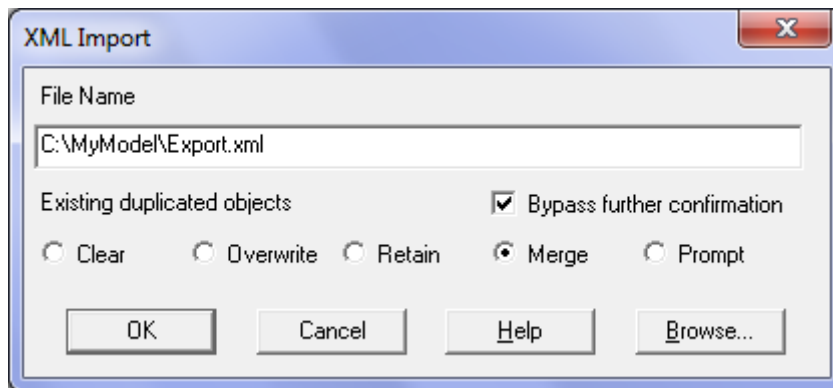
It is important that the PC file have the .xml suffix so that CA Plex will import the file correctly. It is also important that the to file type be Unicode (UTF-8) or ASCII text (Code Page 1208) as all the .xml formatting has already been done by the AS/400 and does not need to be converted. The extended Unicode character set is required as special characters are used in the large properties for some objects.

Import File into CA Plex

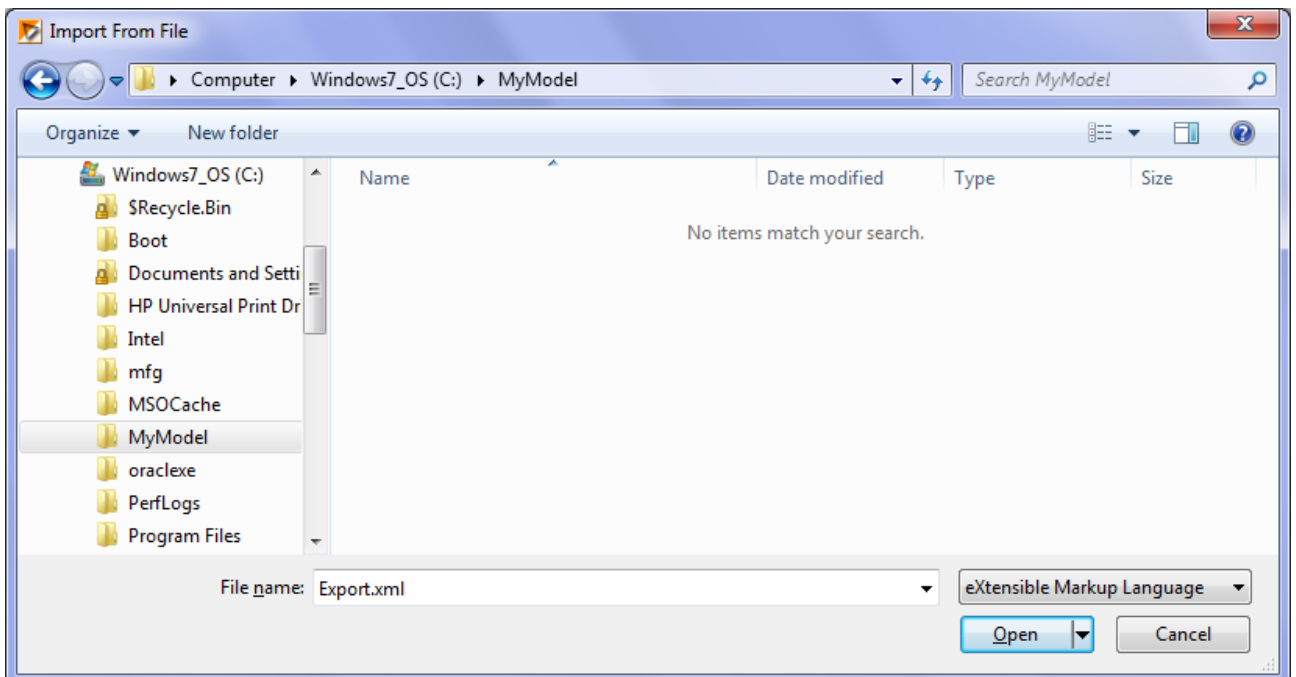
Launch CA Plex.

Follow the instructions in the CA Plex User Guide, Chapter 1 to create your first model. The main steps are to create a group model, add class libraries to the group model, create a local model, and extracting from the class libraries. Once you have saved your local model continue on with the following steps.

Choose menu option Tools/Import/XML Import... to display the XML Import dialog box as displayed below.



Click the Browse... button to display the Import From File dialog box as displayed below.

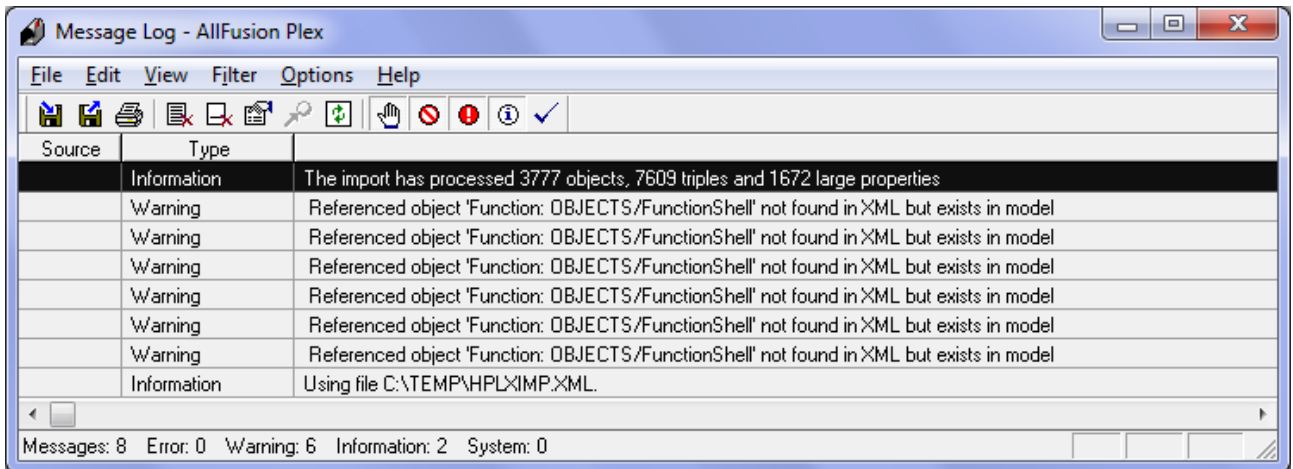


Navigate to the temporary PC directory you transferred the import file into. In the example above this would have been C:\Temp\.

Your import file, in our example HPLXIMP.XML, should now appear in the selection list. Select the import file and click the Open button to return to the XML Import dialog box.

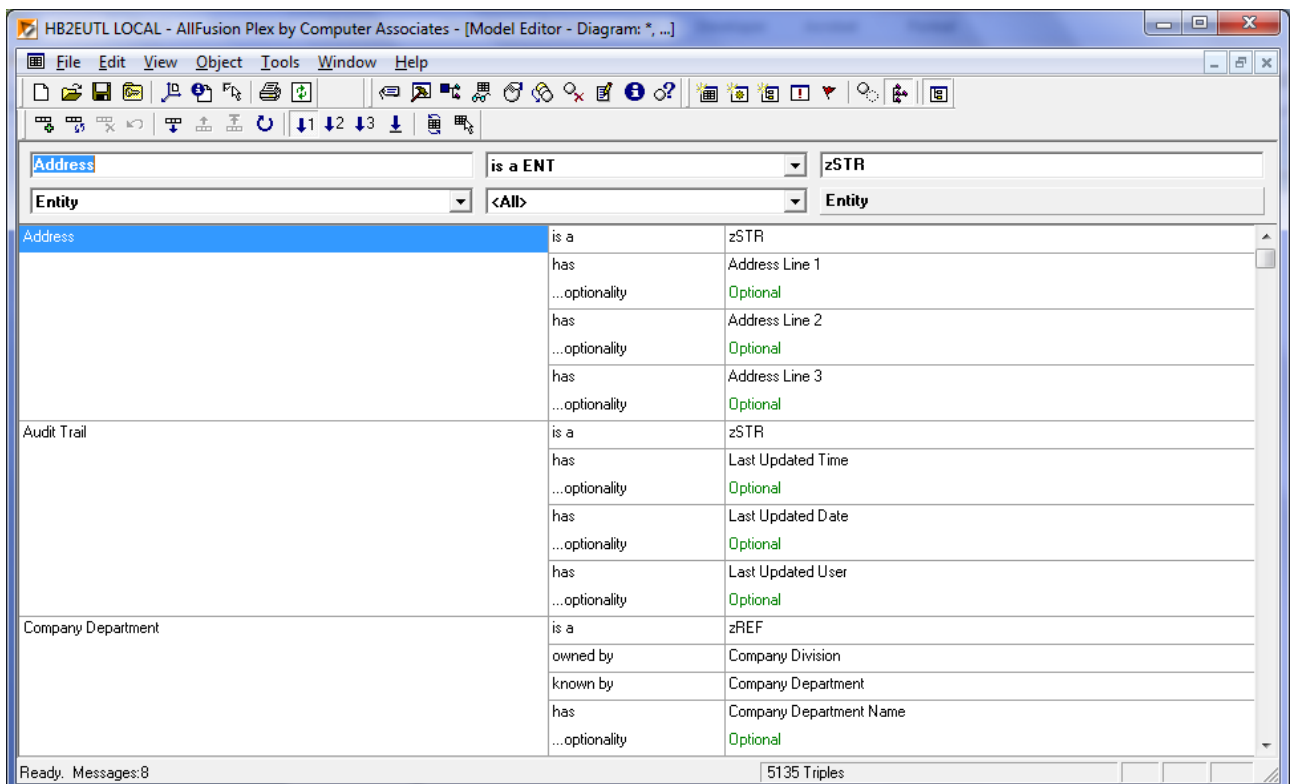
Click the OK button to commence the import of the XML file.

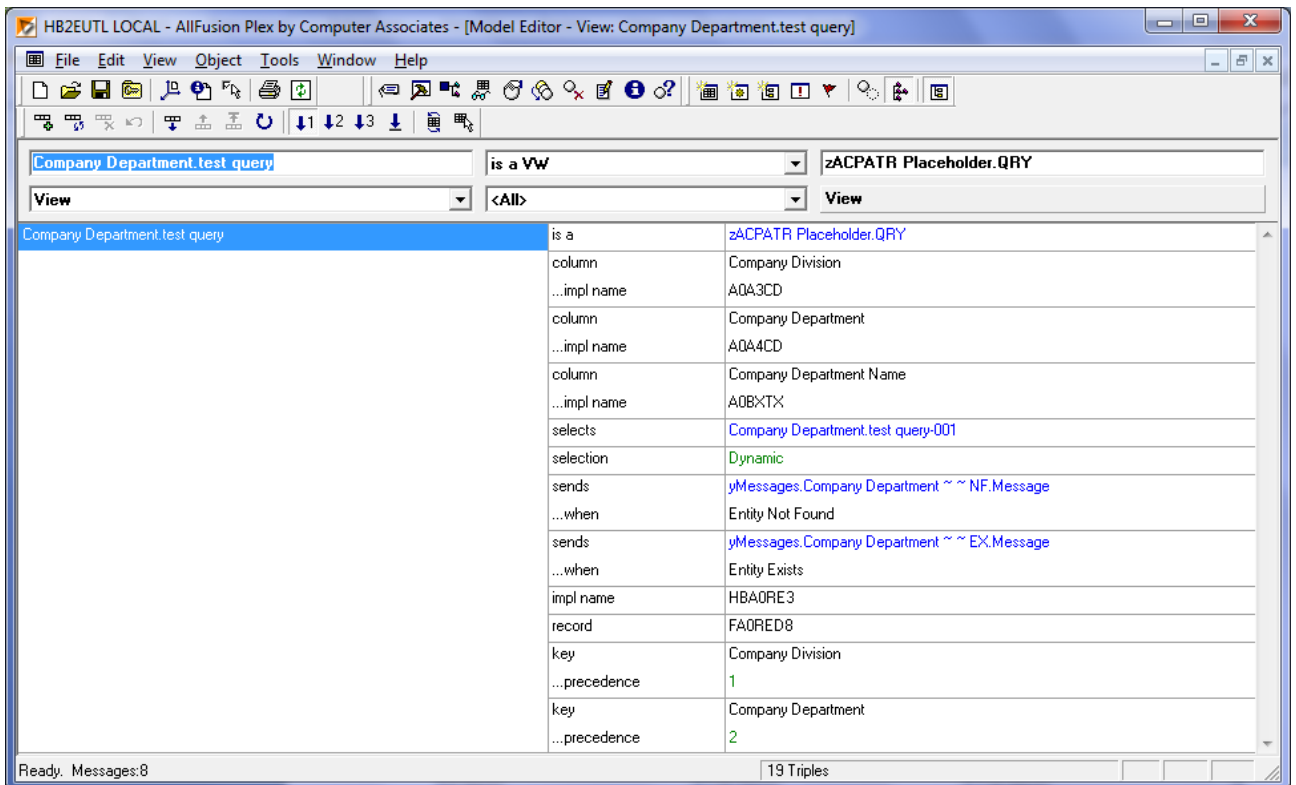
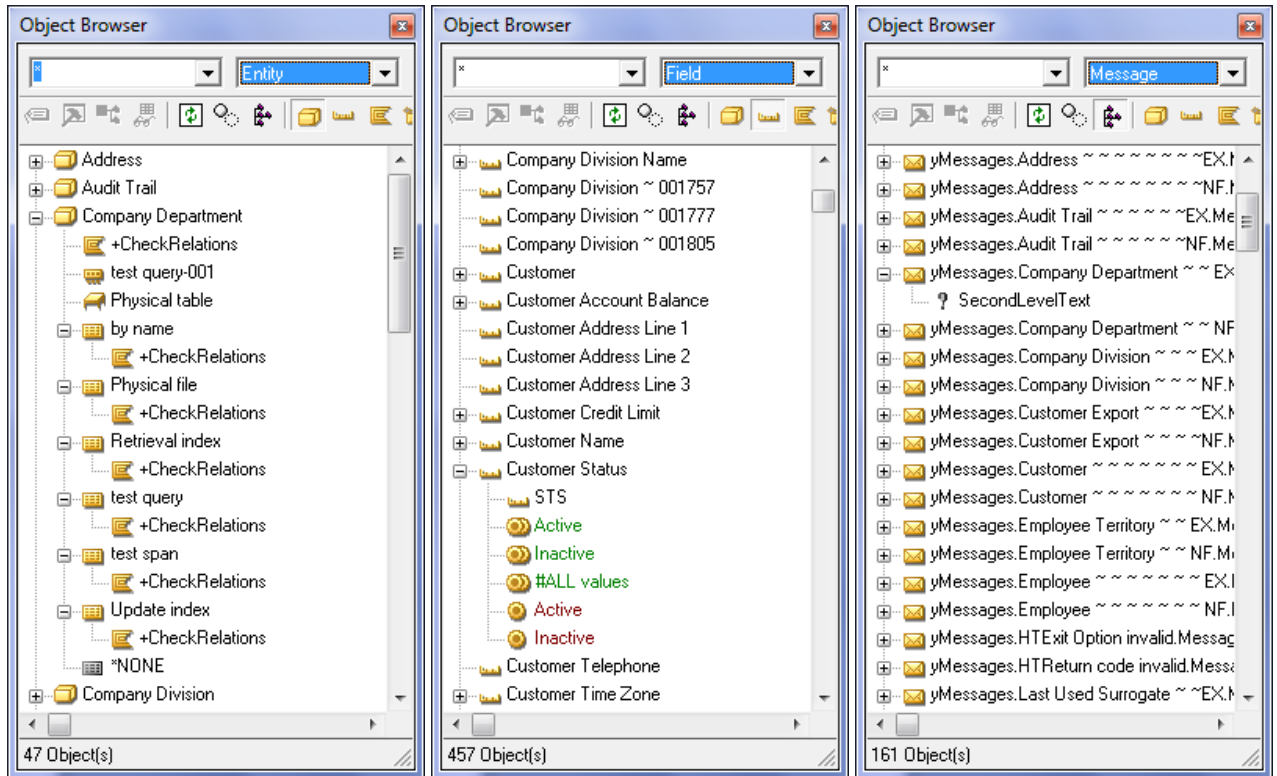
CA Plex may display the Message Log window if any errors have occurred. If not, then you can display the Message Log window by choosing menu option Tools/Message Log.... The following diagram displays a sample Message Log window after a successful import.



If you received an error message then you need to resolve the import errors. If you cannot resolve the errors, then contact HawkBridge for assistance.

Return to the main CA Plex window and press F5 or click the Refresh button to update the display with the information imported. The following diagrams displays a sample Model Editor and Object Browser windows with details imported from CA 2E.





Interface Issues and Design Standard

There were several interface issues with the tool more so to do with CA Plex not supporting certain features of CA 2E data modeling. This section will highlight the most crucial issues and how they have been resolved.

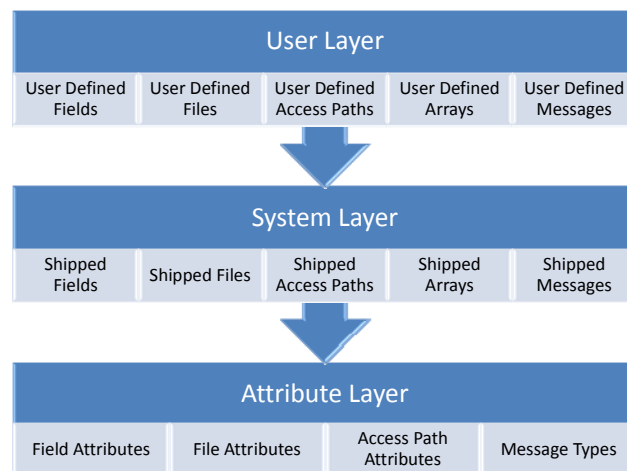
Freedom/Plex Object Type Cross Reference

Freedom/Plex uses the conversions defined in the following diagram to create the export file from CA 2E for use in CA Plex.

CA 2E Object Type	CA Plex Object Type
Field Attribute	Field
File Attribute	Entity and Table
Access Path Type	View
Function Type	Function
Application	Subject Area
File	Entity
Access Path	View
Field	Field
Field Condition	Value(s) and State
Constant	Value and State
Arrays	Entity and View
Function	Function and Source Code
Message	Function, Message and Second Level Text Topic
Narrative	Narrative

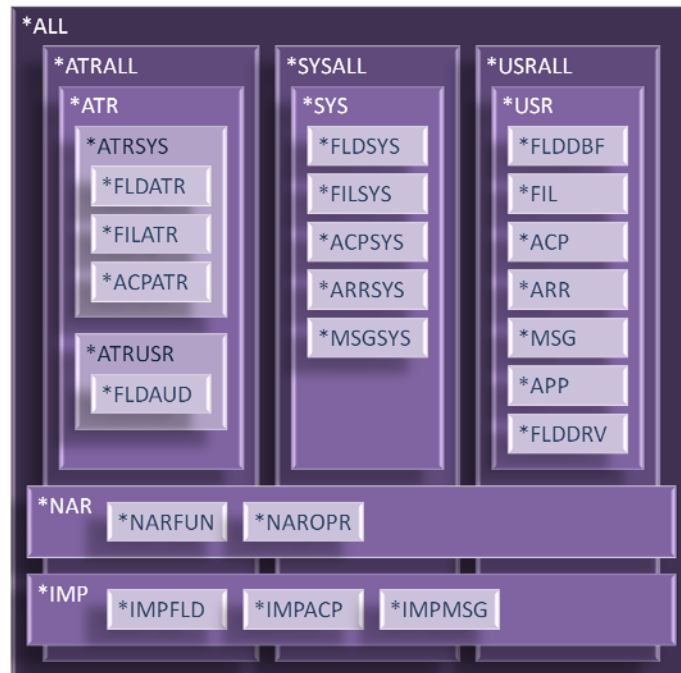
Freedom/Plex Export Layers

Freedom/Plex has been designed with patterns in mind and the ability to extract multiple CA 2E data models into a mature CA Plex environment with multiple layers of patterns. The diagram below depicts the layers of objects as defined within a CA 2E data model that can be extracted in progressive steps using Freedom/Plex.



Freedom/Plex allows you to extract a CA 2E data model in stages so that you can define your CA Plex pattern layers accordingly or even extract components of a layer separately. This is achieved by using the various values for the

extract option on the HCRTPLXERD command. The following diagram illustrates the relationship of the values for the Export options (OPTIONS) parameter. The options are processed from top to bottom and it is possible that a later value may override an earlier option in the list supplied for execution.



CA Plex Attribute and System Layer Object Names

CA 2E system and attribute layer objects that begin with an asterisks (“*”) are exported with a replacement character for the asterisks because CA Plex treats objects commencing with an asterisk in a different manner than CA 2E. Attribute layer objects will have the asterisk replaced with the lower case ‘z’ and system layer objects will be replaced with a lower case ‘y’.

The CA Plex names for attribute and system layer objects have been soft-coded in a shipped file (HPLXOBJ) within the Freedom/Plex library. It is possible for you to edit these names using the Work with Plex Object Names (HWRKPLXOBJ) command. Although the command allows you to specify an alternate template to the default, it is not fully supported and the default template is used to generate the XML for import into CA Plex. This feature may be included in a later release.

Special replacement strings are used in some CA Plex object names in the shipped file (HPLXOBJ) to enable replacement of the attribute and system group model names specified on the Create Plex ERD Import File (HCRTPLXERD) command. The following special replacement strings are used:

- ‘[~:LIB-*ATRLIB-O/]’ is replaced with the value specified on the Attribute group model name (ATRLIBNAM) parameter for the Create Plex ERD Import File (HCRTPLXERD) command, and
- ‘[~:LIB-*SYSLIB-O/]’ is replaced with the value specified on the System group model name (SYSLIBNAM) parameter for the Create Plex ERD Import File (HCRTPLXERD) command.

CA Plex Object Names

CA Plex object naming requirements differ from CA 2E. The following character replacements are performed on CA 2E object names during export to ensure the object names comply with the CA Plex object naming standards:

- Character ‘.’ replaced with ‘:’,
- Character ‘,’ replaced with ‘;’,
- Character ‘/’ replaced with ‘\’,
- Character ‘<’ replaced with ‘[’,
- Character ‘>’ replaced with ‘]’,
- Characters ‘ ’ (double blank) replaced with ‘~’ (single blank and tilde),
- Leading character ‘*’ replaced with ‘#’, and
- Leading character ‘ ’ (blank) replaced with ‘~’.

The replacement characters are soft-coded in a shipped data area (HPLXRPLRFA) within the Freedom/Plex library. The replacement values within the data area can be changed to any valid value that conforms to the CA Plex object naming standards.

CA Plex object names are determined from a number of sources based on the following hierarchy or precedence:

- Object name from the entry in the Plex object name shipped file (HPLXOBJ) using the object surrogate,
- Copy name from the entry in the model object list specified on the Model object list for copy (MDLLST) parameter for the Create Plex ERD Import File (HCRTPLXERD) command using the object surrogate,
- Object name from the entry in the model object list specified on the Model object list for copy (MDLLST) parameter for the Create Plex ERD Import File (HCRTPLXERD) command using the object surrogate,
- Copy name from the entry in the *ALLOBJ model object list using the object surrogate, or
- Object name from the entry in the *ALLOBJ model object list using the object surrogate.

In some cases the object name will be suffixed with the object surrogate in order to avoid a clash in object names during export.

CA 2E Physical Access Paths

CA Plex does not support functions being defined over the table object associated with entities. On the other hand CA 2E allows functions to be based on the Physical file access path. To allow functions based on the Physical file access path to be imported into CA Plex a separate view is created for the Physical file access path. The implementation name for the Physical file view will be suffixed with ‘X’ to avoid an implementation name clash with the table implementation name.

CA 2E Span Access Paths

CA Plex does not support multi-format views and CA 2E span access paths cannot be replicated in CA Plex as they are defined within CA 2E. To resolve this in CA Plex each format of the span index will be generated as a separate view on the associated based on file for the format. If the based on file for the access path is different to the based on file of the format, then the view name will be suffixed with a unique identifier to ensure that an object name clash is avoided. The implementation name for the second view will be suffixed with ‘2’ to also avoid an implementation name clash with the first view.

CA 2E *NONE Access Paths

CA 2E allows *NONE to be specified for the based on access path of a function, and the access path on function parameters. The *NONE view is inherited for all entities from the zCPT entity within the attribute layer to support this feature of CA 2E within CA Plex.

CA 2E Includes Relations

CA Plex does not directly support the Includes relationship as defined within CA 2E. Freedom/Plex will export the Includes relationship as an 'is a' relationship triple within CA Plex.

Although the sequence of relationship triples in the export file are correct, on import into CA Plex they will be re-sequenced so that all inheritance triples are above relationship triples. This has an impact on the sequence of the columns in the associated tables and views which can change the order of parameters on messages and functions. You can force all Includes relations to be exported as 'has' relationship triples within CA Plex by specifying *YES for the Expand Includes relations (EXPINCREL) parameter on the Create Plex ERD Import File (HCRTPLXERD) command.

CA 2E Entry Redirection for Denoted by, Includes, Known by, Qualified by, and Has Relations

CA Plex does not directly support redirection of entries for some relationships as defined within CA 2E. Freedom/Plex will export the entries resulting from a redirection by generating each entry as a 'known by' or 'has' relationship triple within CA Plex.

CA 2E Relationship Checks and Overrides

CA Plex does not directly support the level of relationship checking and overrides allowed within CA 2E data models. Freedom/Plex will generate a meta function call '+CheckRelations' for every entity, view and function to allow the CA 2E relationship checks and overrides to be supported within CA Plex.

For this release of Freedom/Plex the +CheckRelations function only includes functionality for determining the validation view of a relation. A Meta variable, +CheckRelationsValidateView, will be defined with the view that is to be used to validate a specific relation on a file.

Command Reference

Create Plex ERD Import File (HCRTPLXERD)

```

>>--HCRTPLXERD----->
>----->
|      +-*MDLLST-----+
|      +-*M-----+
|      +-*ALLOBJ-----+
|      +-*A-----+
|      +-*SELECT-----+
|      +-*S-----+
|      +-*OBJSGT-----+
|      +-*ATR-----+
|
|      +-*ANY-----+
|      +-*NONE-----+
|      +-*ARRAYS-----+
|      +-*MESSAGES-----+
|
|
|      +-*ALL-----+
|      +-*ACP-----+
|      +-*APP-----+
|      +-*ARR-----+
|      +-*FLD-----+
|      +-*FIL-----+
|      +-*MSG-----+
|
+-OBJNAM(-----object-owner-----object-name-----object-type-----)-+
>----->
|      +-*OBJNAM-----+
+-OBJSGT(-----object-surrogate-----)-+
>----->
|
|      +-HPLXCPYLIST-----+
|      +-*S-----+
|
|      +-*MDLLIB/-----+ +-*SELECT-----+
+-MDLLST(-----library-name/-----model-object-list-name-----)-+
>----->
|      +-*NO--+
+-EDIT(-----*YES-----)-+
>----->
|      +-*GENLIB/-----+ +-HPLXIMPDTA-----+
+-FILE(-----library-name/-----import-file-name-----)-+
>----->
|      +-*FILE-----+
+-MEMBER(-----member-name-----)-+

```

```

>----->
|
|      +-*ALL----+
|      +-*ATRALL-+
|      +-*ATR----+
|      +-*ATRSYS-+
|      +-*ATRUSR-+
|      +-*FLDATR-+
|      +-*FLDAUD-+
|      +-*FILATR-+
|      +-*ACPATR-+
|      +-*MSGTYP-+
|      +-*SYSALL-+
|      +-*SYS----+
|      +-*FLDSYS-+
|      +-*FILSYS-+
|      +-*ACPSYS-+
|      +-*ARRSYS-+
|      +-*MSGSYS-+
|      +-*USRALL-+
|      +-*USR----+
|      +-*FLDDBF-+
|      +-*FIL----+
|      +-*APP----+
|      +-*ACP----+
|      +-*FLDDRV-+
|      +-*ARR----+
|      +-*MSG----+
|      +-*NAR----+
|      +-*NAROPR-+
|      +-*NARFUN-+
|      +-*IMP----+
|      +-*IMPFLD-+
|      +-*IMPACP-+
|      +-*IMPMSG-+
|      +-*EXPINC-+
|
|+-*OPTIONS(----*IMPMSG----)-+
|
>----->
|
|      +-*NONE-----+
|+-*ATRLIBNAM(----attribute-group-model-name----)-+
|
>----->
|
|      +-*NONE-----+
|+-*SYSLIBNAM(----system-group-model-name----)-+
|

```

Purpose

The Create Plex ERD Import File (HCRTPLXERD) command builds an import file of the CA 2E data relationship and data definition model objects for use with CA Plex.

Model object name (OBJNAM)

Specifies the name of the model object to be exported from CA 2E and included in the CA Plex import file.

The model object list specified on the Model object list for copy (MDLLST) parameter will be replaced with the model objects specified on this parameter with the copy flag explicitly specified.

This parameter consists of three elements that together identify a model object.

*MDLLST

A single value indicating that the model object list specified on the Model object list for copy (MDLLST) parameter is used to select the model objects for export to CA Plex.

***M**

A single value indicating that the model object list specified on the Model object list for copy (MDLLST) parameter is used to select the model objects for export to CA Plex.

***ALLOBJ**

A single value indicating that all model objects are to be selected for export to CA Plex.

***A**

A single value indicating that all model objects are to be selected for export to CA Plex.

***SELECT**

A single value indicating that the function is selected using an interactive display function.

***S**

A single value indicating that the function is selected using an interactive display function.

***OBJSGT**

A single value indicating that the model object surrogate specified on the Model object surrogate (OBJSGT) parameter is used to select the model object for export to CA Plex.

***ATR**

A single value indicating that only the attribute layer model objects are to be selected for export to CA Plex.

object-owner

Specify the 25-character name of the model object that owns the object.

***ANY**

Indicates that any model object can be the owner of the object.

***NONE**

Indicates that the model object does not have an owner.

***ARRAYS**

Indicates that the internal file *ARRAYS is the owner of the model object.

***MESSAGES**

Indicates that the internal file *MESSAGES is the owner of the model object.

object-name

Specify the 25-character name of the object.

***ALL**

Indicates that all model objects of the owner are to be selected.

object-type

Specify the 3-character name of the object type.

The possible object type values are:

***ALL**

Indicates that all object types are to be selected.

***ACP**

Indicates that the object to be selected is an access path.

***APP**

Indicates that the object to be selected is an application area.

***ARR**

Indicates that the object to be selected is an array.

***FLD**

Indicates that the object to be selected is a field.

***FIL**

Indicates that the object to be selected is a file.

***MSG**

Indicates that the object to be selected is a message.

Model object surrogate (MDLSGT)

Specifies the internal surrogate of the model object to be exported from CA 2E and included in the CA Plex import file.

***OBJNAM**

A single value indicating that the model object name specified on the Model object name (OBJNAM) parameter is used to select the model object for export to CA Plex.

object-surrogate

Specify the 7-digit surrogate of the object.

Model object list for copy (MDLLST)

Specifies the name and library of the model object list used to select the model objects to be used by the command.

When *MDLLST is specified for the Model object name (OBJNAM) parameter the user must build and specifically select objects using 1=Select from the prompt in the Edit Model Object Copy List (YEDTCPYLST) command. The list is then processed in the same manner as the Copy Model Objects (YCPYMDLOBJ) command.

Only those model objects that are selected for copy on the model object list and fit the selection criteria specified in the Export options (OPTIONS) parameter will be processed by the command.

HPLXCPYLST

Indicates that the default model object list is HPLXCPYLST.

*SELECT

Indicates that the model object list is selected using an interactive display function.

*S

Indicates that the model object list is selected using an interactive display function.

The possible library values are:

*MDLLIB

Indicates that the first data model library found in the current job's library list is used.

model-library-name

Specify the name of data model library.

Edit copy list (EDIT)

Specifies whether or not the model object list is to be displayed in edit copy list mode after it is built and before the extract takes place.

The possible values are:

*NO

Indicates that the model object list is not to be edited.

*YES

Indicates that the model object list is to be edited.

Import file (FILE)

Specifies the name and library of the database file to which the output of the command is directed. If the file does not exist, this command creates a database file in the specified library.

HPLXCPYLST

Indicates that the default import file is HPLXIMPDTA.

The possible library values are:

*GENLIB

Indicates that the library of the database file is the generation library for the data model.

library-name

Specify the library where the file is located.

Import member name (MEMBER)

Specifies the name of the database file member that receives the output of the command. If the member does not exist, this command creates it.

*FILE

Indicates that the name of the member is the same as that of the database file that contains the member specified in the **Import file** (FILE) parameter. If the member does not exist, this command creates it.

Extract options (OPTIONS)

Specifies which options to be applied when selecting and processing the selected model objects for export to CA Plex.

The possible values are:

*ALL

Indicates that all object layers are selected and all details are to be exported. This is equivalent to specifying *ATRALL, *SYSALL and *USRALL.

*ATRALL

Indicates that all attribute layer objects are selected and all details are to be exported. This is equivalent to specifying *ATR, *NAR and *IMP.

*ATR

Indicates that all attribute layer objects are selected to be exported. This is equivalent to specifying *ATRSYS and *ATRUSR.

*ATRSYS

Indicates that all system supplied attribute layer objects are selected to be exported. This is equivalent to specifying *FLDATR, *FILATR, *ACPATR and *MSGTYP.

*ATRUSR

Indicates that all user defined attribute layer objects are selected to be exported. This is equivalent to specifying *FLDAUD.

***FLDATR**

Indicates that all system supplied field attribute layer objects are selected to be exported.

***FLDAUD**

Indicates that all user defined field attribute layer objects are selected to be exported.

***FILATR**

Indicates that all file attribute layer objects are selected to be exported.

***ACPATR**

Indicates that all access path attribute layer objects are selected to be exported.

***MSGTYP**

Indicates that all message attribute layer objects are selected to be exported.

***SYSALL**

Indicates that all system layer objects are selected and all details are to be exported. This is equivalent to specifying *SYS, *NAR and *IMP.

***SYS**

Indicates that all system layer objects are selected to be exported. This is equivalent to specifying *FLDSYS, *FILSYS, *ACPSYS, *ARRSYS and *MSGSYS.

***FLDSYS**

Indicates that all field system layer objects are selected to be exported.

***FILSYS**

Indicates that all file system layer objects are selected to be exported.

***ACPSYS**

Indicates that all access path system layer objects are selected to be exported.

***ARRSYS**

Indicates that all array system layer objects are selected to be exported.

***MSGSYS**

Indicates that all message system layer objects are selected to be exported.

***USRALL**

Indicates that all user defined layer objects are selected and all details are to be exported. This is equivalent to specifying *USR, *NAR and *IMP.

***USR**

Indicates that all user defined layer objects are selected to be exported. This is equivalent to specifying *FLDDBF, *FIL, *ACP, *FLDDRV, *ARR, *MSG and *APP.

***FLDDBF**

Indicates that all database field user defined layer objects are selected to be exported.

***FIL**

Indicates that all file user defined layer objects are selected to be exported.

***ACP**

Indicates that all access path user defined layer objects are selected to be exported.

***FLDDRV**

Indicates that all derived field user defined layer objects are selected to be exported.

***ARR**

Indicates that all array user defined layer objects are selected to be exported.

***MSG**

Indicates that all message user defined layer objects are selected to be exported.

***NAR**

Indicates that all narrative text for selected objects is to be exported. This is equivalent to specifying *NAROPR and *NARFUN.

***NAROPR**

Indicates that operational narrative text for selected objects is to be exported.

***NARFUN**

Indicates that functional narrative text for selected objects is to be exported.

***IMP**

Indicates that all implementation names for selected objects are to be exported. This is equivalent to specifying *NAROPR and *NARFUN.

***IMPFLD**

Indicates that implementation names for selected fields are to be exported.

***IMPACP**

Indicates that implementation names for selected access paths are to be exported.

***IMPMSG**

Indicates that implementation names for selected messages are to be exported.

***EXPINC**

Indicates that all Includes relationships are to be expanded to the associated Has relationships.

Attribute group model name (ATRLIBNAM)

Specifies the name of the CA Plex group model that attribute layer objects are defined in.

***NONE**

Indicates that the attribute layer objects are defined in the same CA Plex model as those being exported.

System group model name (SYSLIBNAM)

Specifies the name of the CA Plex group model that system layer objects are defined in.

***NONE**

Indicates that the system layer objects are defined in the same CA Plex model as those being exported.

**** END OF DOCUMENT ****