SEE/Change

Change Management for the AS/400

Version 4.2

9

Interfaces

Information in this document is subject to change without notice and does not represent a commitment on the part of Thenon Software Pty Limited.

Thenon Software Pty Limited provides no warranty of any kind with regard to this material, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose.

Thenon Software shall not be liable for errors contained herein or for any direct, incidental, or consequential damages resulting from the use of this material.

The software described in this document is furnished under a licence agreement or non-disclosure agreement. The software may be used or copied only in accordance with terms of the agreement. No part of this manual may be reproduced, translated or transmitted in any form or by any means, electronic or mechanical, including photocopying and recording, for any purpose without the express written permission of Thenon Software Pty Limited.

© Copyright Thenon Software Pty Limited, 1993. All rights reserved.

THENON® is a registered trademark of Thenon Software Pty Limited.

IBM®, SYSTEM/38®, AS/400®, OS/400® are registered trademarks of International Business Machines Corporation.

The following are other trade marks of their respective companies: Aspect Computing Pty: LANSA; System Software Associates Inc: AS/SET; Synon Ltd: SYNON/2; RBA Software Ltd: PROJECT MASTER

All rights acknowledged.

Contents

Introduction	
LANSA interface	
Key concepts	
Interface boundaries	
Terminology	
Organisation	
Object development	
Object movements	
Restrictions	9-10
Specifying the controlling parameters	
Configuring applications	9-12
Including LANSA libraries in SEE/Change job descriptions	9-15
Change management using SEE/Change-LANSA interface	
Investigation requests	
Change Requests	
Object development	9-16
Task status areas information	
Ellor codes	
Tachnical Deference	
LAINSA files accessed by SEE/Chalige	0.21
	9-21
ASSET interface	9-23
Key concepts	9-23
Configuration	9-23
In ASSET	
In SEE/Change	
Change management using the SEE/Change-ASSET interface	9-26
Change Requests	9-26
Object development	9-26
What is imported	9-26
Automatic data distribution	9-27
Object locking problems	9-27
	0.00
SYNON/2 interface	
In SYNON/2	9-29
In SEE/Change	9-30
Change management using the SEE/Change-SYNON/2 interface	9-32
Change requests	
Ubject development	
w nat is imported	
Object locking problems	9-33

PROJECT MASTER interface
Key concepts
Entities
Interface menu
Starting the interface
Ending the interface
Transferring SEE/Change data
New Change Requests
Modified Change Requests
Deleted Change Requests
Additional considerations
Transferring PROJECT MASTER data
Command PSTRTHINT: Start the Interface
Command PTFRTHDTA: Transfer SEE/Change data
Command PTFRPMTDTA: Transfer PROJECT MASTER data
Command PENDTHINT: End the Interface
Index

Introduction

This manual documents the interfaces to third party products that are available with SEE/Change. It provides you with a description of the various interfaces, and explains how to get each of these products working in tandem with SEE/Change.

This manual contains supplementary information relating to each interface.

What this manual assumes you already know

This manual should be read in conjunction with the *Configuration Manager, Change Manager, Development Manager* and *Release Manager User and Reference Manuals.*

It is assumed you have a general working knowledge of SEE/Change, and are familiar with its change management cycle.

LANSA interface

SEE/Change facilitates an interface with the LANSA CASE tool, to allow the inclusion of software development using the LANSA product in the SEE/Change change management cycle.

Key concepts

The interface between the two products is based on the implementation of the Task Tracking feature in LANSA release 4 onwards.

SEE/Change accesses the information supplied through Task Tracking, and constructs an external view on the object status within LANSA.

SEE/Change is used as the front-end for development activities within LANSA, and as the back-end for object distribution.

Interface boundaries

Within the context of change management for LANSA based software, the following responsibility boundaries are defined:

SEE/Change is responsible for:

- Investigation and Change Request information
- Object status enquiries
- Initiation of development activities within LANSA
- Initiation of object movements within LANSA
- Release packaging
- Release distribution
- Release installation

LANSA is responsible for:

- Object locking and user access control
- Actual object development
- Actual object movement

Terminology

The term *object* refers to a virtual object as it is known within LANSA, i.e; Field, Process, Function etc. SEE/Change initiates actions on these objects.

Each virtual object is implemented at the end of the development cycle as a set of operating system objects like programs and files. However, the transformation of virtual objects into operating system objects is done wholly within the LANSA product, via export and import routines initiated by SEE/Change.

Organisation

Applications developed in LANSA and managed by SEE/Change are organised as follows:

- For each application environment configured within SEE/Change (*MDL, *ACP, *LIV) a parallel LANSA partition is used.
- An additional development partition is configured within LANSA. All programming activities take place only within that partition. Other partitions (*MDL, *ACP, *LIV) are blocked for development access.
- Each LANSA partition contains all application components. ALL live objects also reside in the acceptance, module and development partitions.
- LANSA objects are promoted only in one direction: *DEV => *MDL => *ACP => *LIV. Objects are not reverted to a previous version. This implies that changes made in the development partition are *pushed* through the change management cycle into the live partition, or refreshed (manually) from the live partition.
- LANSA partition definition allows for the definition of one program library and one database library. Additional database libraries for the Acceptance and Live partitions can be maintained via SEE/Change, provided they are configured in SEE/Change as application database libraries.
- All LANSA virtual objects are considered as application BASE objects within SEE/Change (*BAS). No site/group specific programs are allowed for LANSA objects.

Object development

Object development within LANSA is instigated by SEE/Change with a Change Request (CR) Number. The SEE/Change CR Number is the same as the LANSA Task Identifier. When control is returned, SEE/Change re-constructs the object register (for the CR) based on the current contents of LANSA Task Tracking files.

SEE/Change does not manage, store, archive, or in any way manipulate LANSA source members.



Object movements

Object movements are effected via the LANSA Export and Import routines. The following procedure is used:

- 1. SEE/Change constructs the Export List based on the object register.
- 2. SEE/Change invokes the Export Routine from the development partition to QTEMP (or to a release packet work library).
- 3. SEE/Change constructs the Import List (based on the export list).
- 4. SEE/Change invokes the Import Routine from QTEMP (or from the release packet work library) to the target partition (*MDL, *ACP or *LIV).

Restrictions

There are some restrictions that, at present, must be effected via manual procedures to ensure smooth operation of the interface. The following points should be taken into consideration by all LANSA users:

- Access to LANSA should be restricted to SEE/Change. If LANSA is accessed externally, SEE/Change may not be able to distribute modified objects.
- Once a development session has been invoked through SEE/Change, the current development partition and its default database library should not be changed. It is the user's responsibility to ensure development is carried out only in the development partition known to SEE/Change, using the default database library specified in the LANSA partition definition.
- Distribution patterns within SEE/Change are controlled by the application definition. For an RPG based CR, SEE/Change makes an implicit connection between each object and its owning application. For a LANSA based CR, SEE/Change does not handle object retrieval, and therefore cannot ensure a fixed connection between an object and its owning application. The distribution pattern depends on the CR application. Once LANSA development functions are invoked, any object can be associated with the CR; therefore, it is possible to retrieve an object for modification within the context of one application, and subsequently retrieve it in the context of a different application. It is the user's responsibility to control the application context.
- LANSA virtual object type: *Function*, is qualified by its parent Process, i.e; *Function* name is unique within *Process* name. SEE/Change cannot handle multiple functions with the same name under different process names, i.e; SEE/Change requires all *Function* names to be unique within each partition, regardless of their parent *Process* names. It is the user's responsibility to ensure no duplicate *Function* names are used within any one partition.

Alternatively, users can use the *DIRECT function option by including RDML command FUNCTION OPTIONS(*DIRECT). This will ensure *Function* name is unique within the partition.

• LANSA export/import routines are restricted to 1000 virtual objects (per type) in any one list. This implies the restriction of SEE/Change-LANSA based CRs to 1000 objects (per type).

Configuration

Within SEE/Change, the LANSA interface configuration is specified using:

- *Configuration Manager* function WRKPRMDTA (Work with Parameter Data)
- Configuration Manager function WRKAPPCFG (Work with Application Configuration)

The configuration process allows you to set up applications that allow both LANSA based and RPG based development. You go about the configuration process in the standard way as described in *Configuration Manager User and Reference Manual*.

It is advisable to fully configure SEE/Change for RPG based development, even if you anticipate only LANSA based development - you'll always need the odd RPG based program to complement a LANSA based application.

For applications being developed in LANSA, you provide additional information. This additional information is validated against the partition definition data in LANSA. It is therefore essential you perform the configuration in the following order:

- 1. Plan the configuration layout.
- 2. Specify the partition definitions in LANSA.
- 3. Specify the controlling general parameters in SEE/Change (using function WRKPRMDTA).
- 4. Specify the application details in SEE/Change (using function WRKAPPCFG).
- 5. Change job descriptions to include LANSA libraries in the initial library list.

Specifying the controlling parameters

Menu/Option:	SEECFG / 11. Work with General Parameters
Command:	WRKPRMDTA

The following general parameter codes control the interface:

@INB	LANSA interface active ? Valid values are *YES or *NO. Change this parameter to *YES if you wish to use the interface. *NO will inhibit you from configuring LANSA related information.
@LNS	LANSA program library Name.
@LND	LANSA data library Name.
@LN1	Include file data ? LANSA movement parameter default. Valid values are *YES or *NO.
@LN2	Include compiled objects ? LANSA movement parameter default. Valid values are *YES or *NO.
@LN3	Omit RDML source ? LANSA movement parameter default. Valid values are *YES or *NO.

After specifying the above parameters, you must make the parameter data operational by using:

Menu/Option: SEECFG / 12. Apply General Parameter Changes Command: UPDPRMDTA

Configuring applications

Menu/Option: SEECFG / 2. Work with Application Configuration Command: WRKAPPCFG

The following panel is shown when you use option **2=Change** or **3=Create** from the *Work with Application Configuration* panel.

Default CASE tool The default value is *NONE, i.e; no CASE tool is used. You can prompt for valid values using F4.
 The value specified for this field is used as the default when creating CRs for this application. The value you specify here can be changed when creating a CR via *Change Manager* function WRKCHGRQS (Work with Change Requests).
 If most of the application CRs will be developed using LANSA, select the value of *LANSA.

Continue with application configuration in the normal way.

After you have specified all the standard application configuration details, the panel entitled *Work with LANSA Partition Details* is displayed. Alternatively, this panel is shown when you use option **16=CASE info** from the *Work with Application Configuration* list panel:

THNDEV SEE/Change Testing Environment Work with LANSA Partition Details Application . . . : RC1 RC app LANSA Enter details only against the Systems where the application is used... -----LANSA partition codes-----Type Development Integration Accept/QA Live/Prod Archive System <-- Locate San Frncis Prod DV1 RLV SY1-Asia/P Dev RMD SY2-Europe Prod Thenon Dev Prod Bottom F1=Help F3=Exit F9=Cmd F12=Cancel F24=Messages

This panel enables the entry of LANSA partitions against the application.

As with all other configuration information, you should supply details for all the systems in the network at the primary development centre, and then distribute the configuration data to all systems. You are prompted to enter the Development, Module/Integration, Acceptance/QA, Live/Production and Archive partition codes.

The systems for which you are prompted are the same systems prompted in the panel entitled *Work with Application Program Environments*.

All the partition codes being entered must exist in the LANSA Partition Definition file.

Development	The Development partition code must be entered for a local system that is the development centre for the application.
Module/Integration	The Module/Integration partition code is optional. It can only be entered for a local system that is the development centre for the application. If entered, the following additional rules apply:
	• The Module/Integration program library name (previously configured in the panel entitled <i>Work with Application Program Environments</i>) must be the same as the program library name against the partition code in the LANSA Partition Definition File.

	• The Module/Integration database library name (previously configured in the panel entitled <i>Work with Application Details</i>) must be the same as the database library name against the partition code in the LANSA Partition Definition File.
	If either of the above conditions are not met, a warning message is shown. You can elect to ignore these warning messages, and change the partition definition in LANSA later.
Acceptance/QA	The Acceptance/QA partition code is optional. It can be entered for any system using the application. When entering the acceptance partition code for the local system, the following additional rules apply:
	• The Acceptance/QA program library name (previously configured in the panel entitled <i>Work with Application Program Environments</i>) must be the same as the program library name against the partition code in the LANSA Partition Definition File.
	• At least one of the Acceptance/QA database library names (previously configured in the panel entitled <i>Work with Application Database Details</i>) must be the same as the database library name against the partition code in the LANSA Partition Definition File.
	If either of the above conditions are not met, a warning message is shown. You can elect to ignore these warning messages, and change the partition definition in LANSA later.
Live/Production	The Live partition code must be entered for each system that uses the application. When entering the live partition code for the local system the following additional rules apply:
	• The Live/Production program library name (previously configured in the panel entitled <i>Work with Application Program Environments</i>) must be the same as the program library name against the partition code in the LANSA Partition Definition File.
	• At least one of the Live database library names (previously configured in the panel entitled <i>Work with Application Database Details</i>) must be the same as the database library name against the partition code in the LANSA Partition Definition File.
	If either of the above conditions are not met, a warning message is shown. You can elect to ignore these warning messages, and change the partition definition in LANSA later.
Archive	The Archive partition code is optional. It can be entered for any system using the application. If archive partition is specified, objects replaced in the Live/Production partition are duplicated into the Archive partition before being replaced.

Including LANSA libraries in SEE/Change job descriptions

You must ensure the LANSA libraries are included in the INLLIBL parameter in the following job descriptions:

- SEE/Change's job description OMSJOBD.
- *Communication Manager* job descriptions QMONR, QRCVR, QSNDR and QSRVR at both the development centre and at each remote system.
- Each LANSA application job description (as specified in the Work with Application Details panel).

Change management using SEE/Change-LANSA interface

The standard SEE/Change change management cycle applies to all CRs, including LANSA based CRs. The following pages describe the details associated with LANSA based CRs, and explain the way SEE/Change handles these CRs through the various change management events.

Investigation requests

The change management cycle starts with the raising of an Investigation Request (IR). There are no changes affecting LANSA based development.

Change Requests

When you create a new CR via *Change Manager* function WRKCHGRQS (Work with Change Requests), you are prompted to specify field: **CR CASE Tool**. The default is set to the value you have specified for **Default CASE tool** in the application configuration. You can prompt for other valid values using F4. The value of *LANSA indicates that development is LANSA based. The value of *NONE indicates that the development is RPG based.

Object development

When using Development Manager function WRKCRDEV (Work with CR Development):

- The narrative *LANSA appears to the right of the CR description.
- Action option **31=Crt CR lib** (or function CRTCRLIB) is not valid for LANSA based CRs. CR library is not created.
- Action option **13=Wrk CASE** (or function WRKCASTLS) provides the entry point into a LANSA development session.
- Action option **12=Wrk with CR** (or function WRKCROBJ) is available for both LANSA and RPG based CRs. However, only limited options are available within that function for LANSA based CRs.

When control is returned after a LANSA development session is started (via option **13=Wrk CASE**), SEE/Change attempts to re-construct the object register for the CR, based on information from LANSA Task Tracking files. In order to allow multiple concurrent users to access the same CR, SEE/Change attempts to exclusively lock the CR for the duration of this process. If multiple users return control to SEE/Change at the same time, a situation may occur where SEE/Change is unable to lock the CR for the workstation user. A message will indicate this eventuality, and the SEE/Change object register is not updated. However, no data is lost; the next time the LANSA development session is invoked, the update is re-attempted.

When using function WRKCROBJ (Work with CR objects):

- Object description cannot be changed.
- Source level is always shown as *BAS, i.e; base application.
- Source version is not shown.
- Retrieval status is shown as *LNS.
- Object creation information is not shown.
- Only the following action options are available:

4=Delete20=Movements (or function DSPOBJMVT)21=History (or function DSPOBJHST)

Note that by using option **4=Delete** you remove the reference to that object from the CR, which means that when the CR is moved, the object is not included in the export/import lists. However, if the object has been modified, this might de-synchronise the application partitions.

- Status window option 14. Check CR (or function CHKCR) is not valid.
- LANSA object types are shown as follows:

SEE/Change object type	LANSA object type	Virtual object description
*LNS_FLD	DF	Field definition
*LNS_FILE	FD	File definition
*LNS_PROC	PD	Process definition
*LNS_FUNC	PF	Function definition
*LNS_TEMP	AT	Application template

CR movements

As explained previously, object movements for LANSA based CRs are effected using the LANSA Export/Import routines. SEE/Change invokes these routines in LANSA, then analyses the logs returned and constructs the SEE/Change movement logs, which can be viewed via function DSPOBJMVT (Display Object Movements).

When a movement/promote operation is requested from the main panel of *Change Manager* function WRKCHGRQS (Work with Change Requests), an additional pop up window is shown for all movement/promote types other than Revert to Development, prompting for the following parameters:

Include file data ?	*YES= *NO =	Include data with the exported files. Do not include data with the exported files.
Include compiled objects ?	*YES= *NO =	Exported objects are in compiled form. Exported objects will be re-compiled at import time.
Omit RDML source ?	*YES= *NO =	RDML source is not exported. Program source cannot be re-generated at target systems. RDML source is exported.

These parameters are identical to those that are prompted when creating a manual Export/Import list. Refer to LANSA documentation for further details.

The default values for these parameters can be specified under parameter codes @LN1, @LN2 and @LN3 (refer to *Managing general parameters* in *Configuration Manager User and Reference Manual*). The values you specify on the first movement/promote for the CR are retained as defaults for subsequent movements of the same CR.

Object movements into the Archive partition, if such a partition has been configured for the application, are performed automatically in the context of movement/promote type *LIV. Regardless of the values specified for the *LIV movement/promote, the following values are used in the movements into the Archive partition:

Include file data ?	*NO
Include compiled objects ?	*YES
Omit RDML source ?	*NO

Task status cross-reference

The following table shows the correlation between LANSA Task Tracking status codes, and SEE/Change CR status codes:

LANSA Task Tracking status		SEE/Change CR status	
OPN WRK	Open Work in progress	*DEV	Under development
CLS	Closed	*TST *MDL *ACP *ERM *ERA *ERL	Ready for testing Under Module/Integration testing Under Acceptance/QA testing Errors while moving to *MDL Errors while moving to *ACP Errors while moving to *LIV
FIN Fini	shed / completed	*LIV	In the Live/Production environment

Error codes

The following table shows the error codes that can be generated by SEE/Change when promoting a LANSA based CR:

Error Code	Meaning
*LEW	One or more LANSA export warning was detected
*LEE	One or more LANSA export fatal error was detected
*LIW	One or more LANSA import warning was detected
*LIE	One or more LANSA import fatal error was detected

Error codes *LEE and *LIE will cause the CR status to be in error, and normal recovery is required, i.e; either Revert to Development, or repeat the CR movement/promote that caused the error.

The Revert to Development option does not affect any object deletion for LANSA based CRs. It will only grant development access, and change the completion code of all existing movement logs to *RDV.

Releases

Release Manager function WRKRLS (Work with Releases) performs the following checks at CR allocation time:

• All CRs allocated to the release must share the same CR CASE Tool Id, i.e; you cannot allocate LANSA based CRs and RPG based CRs to the same release, they must be grouped separately.

• Virtual object names must be unique within the release, i.e; you cannot allocate a LANSA based CR containing a virtual object name that is already included in a different CR already allocated to the release.

Technical Reference

LANSA files accessed by SEE/Change

File name	File description
DC@F46	Partition identifier
DC@F75	TTS task definition log
DC@F74	TTS object register
DC@F73	TTS external export / import list entries
DC@F77	TTS external export / import log

Interface cross-reference table

LANSA file	LANSA I/O pgm	SEE/Change pgm	Usage
DC@F46	DC@P8090	OMS620 OMSLNS02	Input Partition code validation Input Pgm / DB library name retrieval
DC@F75	DC@P8093	OMS400 OMS200 OMS250 OMSLNS02	Add Add task record Update Update task details Update Update task status Update Update task status Update Update task status
DC@F74	DC@P8092	OMS200 OMS255 OMSLNS02	Input Refresh SEE/Change object register Update Object de-allocation (RMV) Input Process name retrieval for functions Update Object de-allocation (*LIV)
DC@F73	DC@P8091	OMSLNS02	Update Export / import list construction Delete After export / import runs
DC@F77	DC@P8095	OMSLNS02	Input Export / import completion analysis Delete After export / import runs

ASSET interface

SEE/Change facilitates an interface with ASSET release 2.08 onwards, to allow the inclusion of software development using the ASSET product in the SEE/Change change management cycle.

Key concepts

The object level interface between SEE/Change and ASSET is implemented through the import facility:

- ASSET session is invoked through *Development Manager* function WRKCRDEV (Work with CR Development).
 - When objects are re-generated, their details are recorded in the import file.
 - When control is returned to SEE/Change, the IMPORT function is invoked to register and duplicate objects into the CR environment.
 - After all objects have been successfully imported, change management cycle is as per SEE/Change standards, with special provisions for the distribution of certain data files like help text etc.

Configuration

In ASSET

You must do the following:

- Use ASSET function Edit Application Set to include SEE/Change libraries OMSOBJ and OMSDTA in the set library list.
- Ensure ASSET subsystem job description (ASSETO/ASSETJOBD) includes SEE/Change libraries OMSOBJ and OMSDTA. If ASSET subsystem is active, terminate and re-start the subsystem.
- To inform ASSET that the interface is active, change ASSET data area as follows:

CHGDTAARA DTAARA(ASSETO/ASSETDTA (256 1)) VALUE('Y')

In SEE/Change

You must configure the applications that use ASSET as the primary CASE tool in the development cycle.

Menu/Option: SEECFG / 2. Work with Application Configuration Command: WRKAPPCFG

The following panel is shown when you use option **2=Change** or **3=Create** from the *Work with Application Configuration* panel.

THNDEV	SEE/Change Testing Environment Work with Application Details
Application code Application description Development centre syst Development centre site Application ASP Application job descrip Default CASE tool CASE Model/Set Name . Re-compile objs at remo Re-compile objs at remo Re-compile objs at Dev Distribute source code Distribute source code Distribute object overn Message file operations Multiple versioning ?. Planned concurrent deve Auto revert if promote Number of archiving lev Configuration complexit	<pre>: AP1 : Demo_application_1_* em(P): SY1 SY1-Asia/Pacific (P): SI1 SI1-Hongkong :0 0-16 (0 = use CRTLIB cmd dft) tion: AP1JOBD (P): *ASSET : *NONE te sites ?: N (Y)es/(N)o Centre ? : Y (Y)es/(N)o ?: Y (Y)es/(N)o : D (M)erge/(D)uplicate : Y (Y)es/(N)o lopment ? : Y (Y)es/(N)o errors ? : Y (Y)es/(N)o errors ? : Y (Y)es/(N)o/(P)rompt els: 3 1=Simple 2=Intermediate 3=Complex</pre>
Fl=Help F3=Exit F4=Pr	ompt F9=Cmd F12=Cancel F16=Update F24=Messages

Module/Integ DB Lib	The library where database objects are generated by ASSET, for the set name specified in field CASE Model/Set name .
Default CASE Tool	The defaulted value is *NONE, i.e; no CASE tool is used. You can prompt for valid values using F4.
	The value specified for this field is used as the default when creating CRs for this application. The value you specify here can be changed when creating a CR via <i>Change Manager</i> function WRKCHGRQS (Work with Change Requests).
	If most of the application CRs will be developed using ASSET, select the value of *ASSET.
CASE Model/Set	The set name (in ASSET) that is associated with the application. Every application must be associated with a single set.

Continue with application configuration in the normal way.

Change management using the SEE/Change-ASSET interface

Change Requests

When you create a new CR via *Change Manager* function WRKCHGRQS (Work with Change Requests), you are prompted to specify field: **CR CASE Tool**. The default is set to the value you have specified for **Default CASE tool** in the application configuration. You can prompt for other valid values using F4. The value of *ASSET indicates that development is ASSET based. The value of *NONE indicates that the development is RPG based.

Object development

When using Development Manager function WRKCRDEV (Work with CR Development):

- Action option **13=Wrk CASE** (or function WRKCASTLS) provides the entry point into an ASSET development session. Command INTERAST is prompted for you to select the AUTHOR parameter. The set name is supplied by SEE/Change, based on the application configuration.
- When control is returned from an ASSET development session, SEE/Change attempts to execute the IMPORT routine with MODE(*REGISTER) thereby registering all changed objects under the CR context. Messages will indicate any CR locking problem. After a successful IMPORT run, the objects are registered under the CR, and you can view them using action option 12=Wrk with CR (or function WRKCROBJ), but the actual objects have not been duplicated into the CR library yet.
- When all compilations have ended successfully, you can request a full import run by selecting status window option **35. Import** in the WRKCROBJ panel. Alternatively, you can execute from command entry: IMPORT ... MODE(*FULL). At this point actual objects are duplicated into the CR library.
- The CR cannot be moved to the testing or live environments unless all entries in the import file are fully imported.

What is imported

- The objects you have confirmed for creation in ASSET.
- The Help Text records associated with each interactive program created by ASSET, in files HHT and HXX. These are shown as object type CRDTA entries under the CR.
- The ASSET Help display program CAHELP, which is shown as an object type PGM-NOSRC entry under the CR.
- The ASSET Help display file CAHELPFM, which is shown as an object type DSPF-NOSRC entry under the CR.

Automatic data distribution

- All CRDTA entries under an ASSET CR are automatically distributed at all remote sites after successful CR installation at each production site.
- If you do not need to distribute a CR entry, you can remove it using action option **4=Delete** in the WRKCROBJ panel. When you delete an object in the CR, its corresponding import file entry is also removed.

Object locking problems

Because SEE/Change is informed of the generated objects only after their construction, (as opposed to the standard object retrieval mechanism), CR locks can only be determined when attempting to import these objects. If an object of the same name is already registered under a different CR, a locking problem will occur, and the import run will end unsuccessfully. Two options are available to you to resolve this situation:

- If an import file entry cannot be processed due to the object being already locked to another CR, you should remove it from that other CR, and then re-execute the IMPORT function.
- You can always use status window option **37. Upd imp reg** (or function UPDIMPREG) to remove an existing entry from the import file. Once removed from the import file, the program no longer attempts to import this object.

For more details about the import facility and commands IMPORT and UPDIMPREG, refer to *Development Manager User and Reference Manual.*

SEE/Change facilitates an interface with SYNON/2 release 3.0 onwards, to allow the inclusion of software development using the SYNON/2 product in the SEE/Change change management cycle.

Key concepts

The object level interface between SEE/Change and SYNON/2 is implemented through the Import facility:

- SYNON/2 session is invoked through *Development Manager* function WRKCRDEV (Work with CR Development).
- When objects are regenerated, their details are recorded in the Import File.
- When control is returned to SEE/Change, the IMPORT function is invoked to register and duplicate objects into the CR environment.
- After all objects have been successfully imported, the change management cycle is as per SEE/Change standards, with special provisions for the distribution of certain data files like Help Text etc.

Configuration

In SYNON/2

You must do the following:

• Add a Routing Entry to the batch subsystem being used by SYNON/2 jobs:

ADDRTGE SBSD(...) SEQNBR(...) CMPVAL('THENON') PGM(OMSOBJ/INTERRTG)

- Ensure SYNON/2 Routing Entry with CMPVAL('YCRTOVR') exists in the same subsystem. It will be used by SEE/Change programs.
- Change the job description used by SYNON/2 batch jobs:

CHGJOBD JOBD(...) RTGDTA('THENON') INLLIBL(... OMSOBJ OMSDTA OMSSAV)

• To ensure access to SYNON/2 development sessions is controlled by SEE/Change (i.e; access is denied if an attempt is made to bypass SEE/Change), change SYNON/2 command YEDTMDL to:

CHGCMD(YSY2/YEDTMDL) VLDCKR(OMSOBJ/THENONCHK1)

It is optional. However, if command YEDTMDL is not restricted, change management integrity cannot be guaranteed.

In SEE/Change

First you must specify the SYNON/2 release level in general parameter @SYN within SEE/Change:

Menu/Option:	SEECFG / 11. Work with General Parameters
Command:	WRKPRMDTA

SYNON/2 release level.
 Specify a 3-digit value for the SYNON/2 release you are using, i.e; enter 031 for release 3.1 or 040 for release 4.0 etc.

After specifying the above parameter, you must make the parameter data operational by using:

Menu/Option: SEECFG / 12. Apply General Parameter Changes Command: UPDPRMDTA

Next, you must configure the applications that use SYNON/2 as the primary CASE tool in the development cycle.

Menu/Option: SEECFG / 2. Work with Application Configuration Command: WRKAPPCFG

The following panel is shown when you use option **2=Change** or **3=Create** from the *Work with Application Configuration* panel.

Module/Integ DB Lib	The library where database objects are generated by SYNON/2, for the model name specified in field CASE Model/Set name .
Default CASE Tool	The default value is *NONE, i.e; no CASE tool is used. You can prompt for valid values using F4.
	The value specified for this field is used as the default when creating CRs for this application. The value you specify here can be changed when creating a CR via <i>Change Manager</i> function WRKCHGRQS (Work with Change Requests).
	If most of the application CRs will be developed using SYNON/2, select the value of *SYNON.
CASE Model/Set	The model data library name (in SYNON/2) associated with the application. Every application must be associated with a single model.

Continue with application configuration in the normal way.

Change management using the SEE/Change-SYNON/2 interface

Change requests

When you create a new CR via *Change Manager* function WRKCHGRQS (Work with Change Requests), you are prompted to specify field: **CR CASE Tool**. The default is set to the value you have specified for **Default CASE tool** in the application configuration. You can prompt for other valid values using F4. The value of *SYNON indicates that development is SYNON/2 based. The value of *NONE indicates that the development is RPG based.

Object development

When using Development Manager function WRKCRDEV (Work with CR Development):

- Action option 13=Wrk CASE (or function WRKCASTLS) provides the entry point into a SYNON/2 development session. Command YEDTMDL is prompted for you to select the USER(*DSNR/*PGMR) parameter. The Job List parameter is supplied by SEE/Change, based on application configuration and the CR number.
- When control is returned from a SYNON/2 development session, SEE/Change attempts to execute the IMPORT routine with MODE(*REGISTER) thereby registering all changed objects under the CR context. Messages will indicate any CR locking problem. After a successful IMPORT run, the objects are registered under the CR, and you can view them using action option **12=Wrk with CR** (or function WRKCROBJ), but the actual objects have not been duplicated into the CR library yet.
- When all compilations have ended successfully, you can request a full import run by selecting status window option **35. Import** in WRKCROBJ panel. Alternatively, you can execute from command entry: IMPORT ... MODE(*FULL). At this point actual objects are duplicated into the CR library.
- The CR cannot be moved to the testing or live environments unless all entries in the import file are fully imported.

What is imported

- The objects you have confirmed for creation using the SYNON/2 Job List panel.
- The application message file, if it exists.
- The SYNON/2 Condition File and associated logical views xxVLLSP, xxVLLSL0 and xxVLLSL9 if they exist. They are shown as object type PF-NOSRC and LF-NOSRC entries under the CR.
- The SYNON/2 Condition Display program xxVLLSR, if it exists. It is shown as an object type PGM-NOSRC entry under the CR.
- The SYNON/2 Condition Display File xxVLLSR#, if it exists. It is shown as an object type DSPF-NOSRC entry under the CR.

• The actual Condition File data is copied into a file named @SYNONDTA and is shown as an object type CRDTA entry under the CR.

@SYNONDTA is automatically refreshed within the CR environment, i.e; the contents of file xxVLLSP are copied into file @SYNONDTA replacing any existing data. This ensures that the Condition File being distributed contains the latest data. The refresh operations are contained within function CHKCR, which can be executed by using:

- Status option **14. Check CR** in function WRKCROBJ (Work with CR objects)
- Exit option **2. Exit Rdy Tst** in function WRKCROBJ (Work with CR objects)
- Action option **14=Chg CR *TST** in function WRKCRDEV (Work with CR Development)
- Action option **11=Move/promote** or **14=Chg CR *RDY** against a CR in the status of *TST in function WRKCHGRQS (Work with Change Requests)
- SYNON/2 allows you to generate Help Text as either text member (*TM) or as a panel group object (*UIM). If a text member is generated, the member is imported into file name QTXTSRC in the CR library shown as object type CRDTA, and is automatically distributed at remote systems. If a panel group object is generated, the panel group object is treated as an independent object, and is delivered into the target environment like any other object.

Automatic data distribution

- All CRDTA entries under a SYNON/2 CR are automatically distributed at all remote sites after successful CR installation.
- You need to distribute the Condition Files (xxVLLSP/Lx) only when you distribute an application for the first time. You need to distribute @SYNONDTA whenever changes are made to the Condition File data.
- If you do not need to distribute a CR entry, you can remove it using action option **4=Delete** from WRKCROBJ panel. The corresponding import file entry will also be removed.

Object locking problems

Because SEE/Change is informed of the generated objects only after their construction, (as opposed to the standard object retrieval mechanism), CR locks can only be determined when attempting to import these object. If an object of the same name is already registered under a different CR, a locking problem will occur, and the import run will end unsuccessfully. Two options are available to you to resolve this situation:

- If an import file entry cannot be processed due to the object being already locked to another CR, you should remove it from that other CR, and then re-execute the IMPORT function.
- You can always use status window option **37. Upd imp reg** (or function UPDIMPREG) to remove an existing entry from the import file. Once removed from the import file, the program will no longer attempt to import this object.

For more details about the import facility and commands IMPORT and UPDIMPREG, refer to *Development Manager User and Reference Manual*.

PROJECT MASTER interface

SEE/Change facilitates an interface with PROJECT MASTER to allow the incorporation of project management functionality in the SEE/Change change management cycle.

Key concepts

The interface between SEE/Change and PROJECT MASTER transfers CRs out of SEE/Change; and actual work and costs recorded against them in PROJECT MASTER back into SEE/Change.

You can transfer data between the two systems as often as you wish, and the procedure can be automated, for example, so it is run on a daily basis.

When you first start the interface, it will examine your existing SEE/Change data and transfer all the outstanding CRs into PROJECT MASTER for you. So even if you have been using SEE/Change for some time before implementing the bridge to PROJECT MASTER, you are not faced with a large re-keying exercise.

Entities

The following table shows the relationship between SEE/Change and PROJECT MASTER entities:

SEE/Change		PROJECT MASTER	
File	Entity	File	Entity
XSY	CR originating system	DOMAIN	Domain
XAP	CR application	PROJCT	Project
XIR	Investigation Request (IR)	FUNCTN	Function
XCR	Change Request (CR)	TASK	Task

Interface menu

Menu SEEPMT is dedicated to PROJECT MASTER interface activities:

Secondary menu: SEEPMT

```
SEEPMT THENON/SEE Change Manager
Select one of the following:
    . Start Project Master Interface
    . Transfer Project Master data to THENON
    . End Project Master Interface
Selection or command
===>
F3=Exit F4=Prompt F9=Retrieve F12=Cancel
F13=User support F16=System main menu
```

Each option has its own PROJECT MASTER command. They are:

1. Start Project Master Interface (PSTRTHINT command). You only need to take this option once; it sets up the operating environment for you.

2. Transfer SEE/Change data to Project Master (PTFRTHDTA command). This is the option that transfers new and amended CRs, so typically it is run on a daily basis.

3. Transfer Project Master data to SEE/Change (PTFRPMTDTA command). This option accumulates the actual cost and time spent on each CR that has been transferred to PROJECT MASTER. You may want to run this option on a weekly basis, or more frequently if you need tighter control.

4. End Project Master Interface (PENDTHINT command). You would take this option if you wish to suspend the recording of SEE/Change transactions.

Starting the interface

When you start the interface, a library called PTHENON is created to contain the interface transactions. A journal and journal receiver are created within this library, and journalling of SEE/Change CR file (XCR) commences.

This journal records every transaction entered into the CR file - additions, modifications, and deletions - regardless of how those transactions are created.

This option can only be taken when SEE/Change is not in use elsewhere, as it requires exclusive locks on SEE/Change objects.

Ending the interface

This option stops journalling the CR file, and deletes the journal, cross-reference file and library PTHENON. This option can only be taken when SEE/Change is not in use elsewhere, as it requires exclusive locks on SEE/Change objects.

Transferring SEE/Change data

The transactions transferred are changes to the CR file. Each CR will become a task. Some of the transactions will require changes to other PROJECT MASTER files.

For example, a SEE/Change application is equivalent to a PROJECT MASTER Project. If a CR application does not exist within PROJECT MASTER, a new project is created. In the same way, if the CR originating system does not exist within PROJECT MASTER, a new Domain is created; and if the IR number does not exist within PROJECT MASTER, a new Function is created.

New Change Requests

When a new domain or project needs to be created, the name and description of the equivalent system and application are transferred.

When a new function is added, the name, description and priority of the equivalent IR are transferred.

At the CR level, the following fields are transferred:

- CR sequence number becomes the task name.
- CR text becomes the task description.
- CR estimated hours become the original estimate and estimate to complete for the task.
- CR priority becomes the task priority.
- CR user is transferred if possible, becoming the resource allocated to the task.
- Related IR, application and originating system become related function, project and domain.

In addition, for every item created in PROJECT MASTER, some comments are added to indicate that they were created by the interface, when they were created, who ran the interface and so on.

In SEE/Change, the user assigned to the CR is indicated by entering their AS/400 user profile name. In order to transfer these assignments to PROJECT MASTER, you must first use the *Work with Authorised Users* option in PROJECT MASTER. For each resource you have defined in PROJECT MASTER, enter their user profile name. When the interface runs, it will refer to the authorised users list in order to translate the user profile name into the PROJECT MASTER resource initials. If a user profile is encountered that has no counterpart in PROJECT MASTER, the new task is left unassigned.

For every task created, a cross-reference record is written. This stores the internal numeric identifier of the task and the CR number. This allows you to move tasks around at will within PROJECT MASTER - moving them to different projects, renaming them and so on - and yet the interface still knows which task relates to which CR. So, future modifications can be reflected in the equivalent task, wherever it is, and you don't end up with duplicate tasks even if you radically amend the structure of your projects within PROJECT MASTER.

As an example, you may have a single application in SEE/Change that you wish to organise as two projects in PROJECT MASTER, perhaps with one project for development and one project for maintenance. The interface will initially transfer all tasks for one application into one project in PROJECT MASTER, but it is a simple task for you to globally move all maintenance tasks into a different project. Once moved, modifications will appear in the appropriate project.

Modified Change Requests

Many amendments to CRs will simply be ignored, because the modification has no significance to PROJECT MASTER. However, all changes to the following fields are transferred:

- IR number
- CR sequence number
- CR text
- CR hours
- CR user
- CR priority

Note, that a change to CR hours results in a change to the original estimate for the task, but not the estimate to complete. This is because the estimate to complete is automatically revised as work progresses within PROJECT MASTER.

Deleted Change Requests

When a CR is deleted, the equivalent task is closed. The related function, project and domain are unaffected.

Additional considerations

Before you start the interface, you should set up your resources in PROJECT MASTER if you have not already done so. First, use the *Work with Resources* option to create the resources and their calendars. Then use the *Work with Authorised Users* option to indicate the related user profile name for each resource. By doing this you can ensure that as CRs are transferred, they are assigned (and therefore ultimately scheduled) in PROJECT MASTER.

It is possible to re-start the interface. When the interface is started, it first checks to see if there are any outstanding transactions that have not been processed. If there are, these are transferred first. Even in the event of a power failure, the interface can be re-submitted and it will carry on from where it left off. The corollary of this is that each transaction processed is removed, so it cannot be processed twice by mistake.

A report is produced showing transactions processed, actions taken, problems encountered etc.

Transferring PROJECT MASTER data

This is a straightforward facility to return the progress data captured by PROJECT MASTER to SEE/Change. For each CR that has been transferred to PROJECT MASTER, this option accumulates the total time spent and the total costs to date, and puts these into the actual hours and actual cost fields on the CR. The actual cost is based on PROJECT MASTER calculations, which can allow for multiple different rates, i.e; allow for overtime or special customer rates, and even non-labour costs like travelling and accommodation costs.

Command PSTRTHINT: Start the Interface

This command sets up the environment required by the interface.

Customer name (CUSTMR)

In PROJECT MASTER, one of the things used to specify costs (and charge-outs) is the customer that the work is being done for. Hourly rates can vary from one customer to another. You can specify here which customer should be recorded as a default on tasks created by the interface start-up. Valid values are:

*NONE

No customer is recorded against new tasks.

*SEL

The customer selected from a list prompt is recorded against each new task created.

Data Library (THLIB)

Specifies the SEE/Change data library name. The default is OMSDTA.

Job Description (JOBD)

Specifies the job description used for submitted jobs. The default is PMJOBD.

Command PTFRTHDTA: Transfer SEE/Change data

This command transfers CR transactions to PROJECT MASTER.

Customer Name (CUSTMR)

In PROJECT MASTER, one of the things used to specify costs (and charge-outs) is the customer that the work is being done for. Hourly rates can vary from one customer to another. You can specify here which customer should be recorded as a default on tasks created by the interface start-up. Valid values are:

*NONE

No customer is recorded against new tasks.

*SEL

The customer selected from a list prompt is recorded against each new task created.

Data Library (THLIB)

Specifies the SEE/Change data library name. The default is OMSDTA.

Job Description (JOBD)

Specifies the job description used for submitted jobs. The default is PMJOBD.

Command PTFRPMTDTA: Transfer PROJECT MASTER data

This command transfers progress data from PROJECT MASTER to SEE/Change.

There are no parameters for this command.

Command PENDTHINT: End the Interface

This command stops the SEE/Change-PROJECT MASTER interface and deletes all objects created by the interface. Tasks already transferred to PROJECT MASTER and actuals already transferred to SEE/Change are unaffected.

There are no parameters for this command.

Index

@SYN parameter 9-30 @SYNONDTA file 9-33

Α

ASSET configuration 9-23 CR development 9-26 data area 9-23 importing 9-23, 9-26 library list entries 9-23 object distribution 9-27 subsystem 9-23

C CR

application context 9-10 ASSET based 9-26 creating 9-38 deleting 9-39 description 9-38 hours & cost 9-38, 9-39 journalling 9-37 LANSA based 9-16 locks 9-16, 9-26, 9-27, 9-33 movement log 9-18 number 9-8 object history 9-17 object movements 9-17, 9-18 object register 9-8, 9-9 priority 9-38 **PROJECT MASTER based 9-38** sequence number 9-38 status codes 9-19 SYNON/2 based 9-32 updating 9-39

D

Deleting CR 9-39 journal 9-38 objects 9-17, 9-27, 9-33 Displaying object history 9-17

Ε

Error exporting 9-19 importing 9-19 message 9-16, 9-26, 9-32 recovery 9-19 warning 9-14, 9-19 Error codes 9-19 Exporting error 9-19 list 9-9, 9-17, 9-21 routine 9-9, 9-18

G

General parameters 9-11

I

Importing error 9-19 list 9-9, 9-17, 9-21 routine 9-9, 9-18 Interface ASSET 9-23 LANSA 9-7 PROJECT MASTER 9-35 SYNON/2 9-29 IR application 9-38 definition 9-16 number 9-38, 9-39 priority 9-38

L

LANSA access restrictions 9-10 configuration 9-11, 9-12 CR CASE tool 9-16 CR development 9-16 cross-reference table 9-21 development access 9-8 error codes 9-19 exporting 9-9 function 9-10 importing 9-9 IR definition 9-16 library list entries 9-15 object movement 9-7, 9-9 object restrictions 9-10 object types 9-17 parameters 9-11 partition definition 9-8, 9-13 process 9-10 release 9-19 task identifier 9-8 task tracking files 9-8, 9-16 task tracking status codes 9-19 virtual object 9-7

Μ

Movement logs 9-18, 9-19

0

Object deleting 9-17, 9-27, 9-33 development 9-8, 9-16, 9-26, 9-32 distribution 9-27, 9-33 history 9-17 locks 9-37, 9-38 movements 9-9, 9-17, 9-18

Ρ

Parameters 9-18 PROJECT MASTER commands 9-36 data transfer 9-35, 9-38, 9-39, 9-43 domain 9-38 end interface 9-38, 9-43 function 9-38 hours & cost 9-35 journalling 9-36 menu 9-36 project 9-38 relationship with SEE/Change entities 9-35 resources 9-39 restarting interface 9-39 starting interface 9-36, 9-41 tasks 9-38 user profile 9-38

R

Release CR allocation 9-19 work library 9-9

S

SEE/Change parameters 9-11 SYNON/2 applications 9-30 condition file 9-33 configuration 9-29 CR CASE tool 9-32 CR development 9-32 importing 9-29, 9-32 object distribution 9-33 parameters 9-30 routing entry 9-29