

Demo
Installation
Quick introduction

Relative to Release 90.6

Copyright © 2019 ITSforSURE

All rights reserved. No part of this publication may be reproduced, transmitted, transcribed, stored in a retrieval system, or translated in any form or by any means without prior written consent of the author.

Any product and related material disclosed herein are only furnished pursuant and subject to the terms and conditions of a duly executed Software License Agreement.

ITSforSURE explicitly excludes any liability for all damage, originating from whatever reason, including any direct or indirect damage, such as consequential damage or working damage, which, among other reasons, was caused by the use of the information in this document or software material. There are no warranties extended or granted by this document. The only warranties made by ITSforSURE, if any, with respect to the product described in this document are set forth in the Software License Agreement.

Information contained in this document is subject to change without notice. Revisions may be issued to advice of such changes and/or additions.

Comments or suggestions regarding this document are welcome and should be directed to:

ITSforSURE
Email sure@itsforsure.nl

Any reference to registered trademarks in this document without further explanation of the owners of the respective trademarks implies that all rights to these trademarks remain with their respective owners.

Content

| | | |
|----------|--|-----------|
| 1 | Introduction | 4 |
| 2 | Installation | 5 |
| 2.1 | Installation on the mainframe | 6 |
| 2.1.1 | Copy and install the mainframe software | 6 |
| 2.1.2 | COMS Definitions | 11 |
| 2.1.3 | Run the On-line Program (terminal emulation) | 13 |
| 2.2 | Installation of SURE on the PC | 14 |
| 2.2.1 | Copy and install the PC-software | 14 |
| 2.2.2 | Run SURE | 17 |
| 3 | Options, Parameters and Securities | 18 |
| 3.1 | Define an employee-function | 18 |
| 3.2 | Define a personal usercode with authorizations | 20 |
| 3.3 | Environments | 24 |
| 3.3.1 | Verify the definition of environment DEVELOP | 25 |
| 3.3.2 | Define two new environments: ACCEPTANCE and PRODUCTION | 26 |
| 3.3.3 | Rename environments to your own needs | 28 |
| 3.4 | Define an application system | 30 |
| 3.5 | Various task options | 33 |
| 3.5.1 | Priorities | 33 |
| 3.5.2 | Task Types | 33 |
| 3.5.3 | Task Groups | 34 |
| 3.6 | File Types | 35 |
| 3.7 | File References | 35 |
| 3.8 | Other repository options | 36 |
| 3.9 | Local SURE options | 37 |
| 3.10 | Options verification | 37 |
| 4 | Task registration | 38 |
| 4.1 | Task Registration via SURE | 38 |
| 5 | Load a directory with files in SURE | 41 |
| 6 | Build Support & Distribution Support | 45 |
| 6.1 | Compilation and Object files | 45 |
| 6.2 | Load object run statistics in SURE | 49 |
| 6.3 | Other batch procedures and backup procedures | 49 |
| 7 | Program life cycle support | 51 |
| 7.1 | Modify an existing file | 51 |
| 7.2 | Transfer modifications to the next environment | 53 |
| 7.3 | Add a new file and modify an existing file | 55 |
| 7.4 | Inquire and browse functions | 57 |
| 8 | Support functions | 59 |
| 8.1 | Cross reference and data dictionary functions | 59 |
| 8.2 | Delta files | 60 |
| 8.3 | Find and replace utilities | 61 |
| 9 | Epilogue | 63 |

1 Introduction

Welcome to the SURE 'change & control' system demonstration. We are pleased that you have chosen to investigate how SURE can assist you in your change control needs.

Originally SURE was designed to keep track of changes and to prevent the loss of changes when modifying source modules. The first version was released in 1983 and was specifically tailored for Unisys A-Series systems. From that time, SURE has proven to be a powerful assistant for project leaders, programmers, change controllers, security managers, auditors, and even operators.

Actually, hundreds of data processing professionals in dozens of companies have contributed to the development of SURE by telling us what they needed to perform their duties. The diversity of SURE's existing users assures that the system provides you with immediate and tangible benefits.

As in many other areas of endeavor, a few moments spent initially on understanding the overall picture of the SURE system will be more than repaid when you begin to use SURE in your own environment. This demonstration is designed to guide you from a point where you lack any specific knowledge of SURE to a point where you have seen some of the things SURE can accomplish for you.


During this demonstration, a small SURE database (INFDB) is placed on your system. The demonstration handles the following issues:


- Environments
- Security
- Task registration
- Program life cycle support
- Change control functions
- Auditor functions
- Project leader functions

This demonstration makes a clear distinction between the above mentioned functions, as if each function is performed by a separate person. It is, however, obvious that all functions can be performed by one single person.

Section 2 of the demo is a step by step installation procedure.

Sections 3 and further contain a quick introduction to some SURE features.

Some paragraphs in this document are marked with character  or *.

Character  means that this paragraph contains important information.

Character * means that one or more actions have to be performed.

Please take a few moments to look over the following sections of the SURE demonstration. We appreciate your interest in SURE and hope that by the time you complete this demonstration you will share our enthusiasm for SURE. If we can be of service to you please contact us as follows:

ITSforSURE
Haarlem
The Netherlands

Email: sure@itsforsure.nl

2 Installation

The SURE software is placed on the SURE website www.itsforsure.nl

- * Download the software and unzip:
 - Click on <download> to enter the download section.
 - Download file "MCP and Windows software incl. patches" of the newest release.
 - Unzip the downloaded zip file to a place anywhere on your workstation. You will see the following directory structure:
 - .. SureReleasexx (where xx is a two-digit release number)
 - \CPNX_software
 - \DATA Text Integration
 - \Ris for Windows
 - \SURE for windows
 - \<about 20 files>
 - \Install.exe
 - \Installation of SURE.doc
- * Installation of server software on the mainframe:
 - Directory CPNX_software contains a containerfile SURExxMCPyyy with MCP software (where xx is the SURE release number and yyy is the MCP release number)
 - FileTransfer this containerfile with BINARY FTP as follows
 - o Open command prompt
 - o FTP <your host>
 - o Usercode/password
 - o type binary
 - o put SURExxMCPyyy
 - On the mainframe: install the software via the procedure as described in paragraph "Copy and install the mainframe software".
 - Chapter 9.2 of document "installation of SURE.doc" gives valuable background information about the mainframe software.
- * Installation of client software on a PC:
 - Install the SURE client software via the install.exe program as described in paragraph "Installation of SURE on the PC".
 - Chapter 9.1 of document "installation of SURE.doc" gives valuable background information about the client software.

2.1 Installation on the mainframe

2.1.1 Copy and install the mainframe software

The first step in the SURE demonstration is to copy the SURE software from the tape to your system. It is helpful to read the installation document before you continue with the installation.

During the demonstration, a small SURE database (called INFDB) is placed on your system. This database is the SURE repository.

A SURE repository is subdivided into logical environments. Each environment corresponds with a phase in the development process. The amount of logical environments and the names of these environments are normally determined by the customer (= you). Common environment names are DEVELOP, DESIGN, ACCEPTANCE, INTEGRATION, PRODUCTION, HISTORY et cetera.

This demonstration starts with a repository that has only one environment: DEVELOP. During the demonstration, two extra environments are created: ACCEPTANCE and PRODUCTION. For each environment you need at least two usercodes:

- A usercode where a user logs on to do this jobs (programming etc.). This is the WORK usercode.
- A usercode where the SURE evening batch runs. This is the BATCH usercode.

Since you also need a separate installation usercode and a personal usercode for authentication in SURE, the total amount of new usercodes that you need for this demo is eight.

SURE can be installed in any directory on your system. You can choose your own usercode and diskpack where SURE must be installed. During this demo we use the following usercodes:

| Usercode | | Purpose |
|---------------------|----|---|
| SURE | PU | Installation usercode where the software must be installed. |
| <personal usercode> | | for authentication, for example a persons name like SIMON |
| DEV | | Work usercode for environment DEVELOP. |
| DEVBATCH | PU | Batch usercode for environment DEVELOP. |
| ACC | | Work usercode for environment ACCEPTANCE. |
| ACCBATCH | PU | Batch usercode for environment ACCEPTANCE. |
| PRO | | Work usercode for environment PRODUCTION. |
| PROBATCH | PU | Batch usercode for environment PRODUCTION. |

You are free to use your own set of usercodes. In that case you must substitute the above mentioned usercodes with your own usercodes.

All usercodes must have the same FAMILY specification. This family specification can be decided by you.

The family specification used in this document is FAMILY DISK = DEMOPK OTHERWISE DISK.



Please make sure that the installation usercode (SURE) is a privileged user (PU), otherwise the installation will fail.

The installation can now be started.

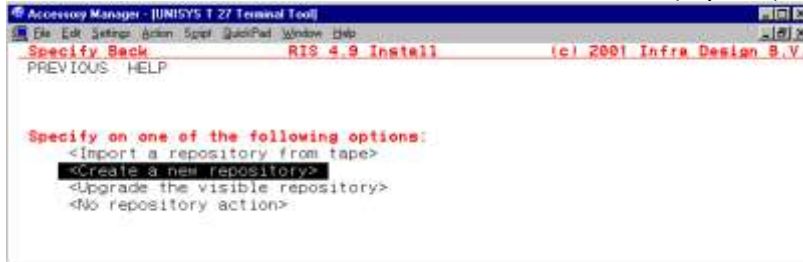
- * The software is delivered via a wrapped container file. Unwrap the container file as follows:

1. Log on to Cande with the usercode where the file is placed.
2. Unwrap the container file:

- WFL UNWRAP *= OUTOF <container> TO <pack> (PACK, RESTRICTED=FALSE)
 - This copies the software to usercode-directory (INFRA) on <pack>
3. Log on via Cande with the installation usercode (SURE):
 - HELLO SURE/<password>
 4. Process the installation DO-file:
 - DO (INFRA)RESPECT/SETUP(" <packname>(PACK)")

Set-up copies additional files to the installation usercode and initiates the installation program called OBJECT/RIS/INSTALL.

The 1st installation screen: Define the installation action: new, update, import



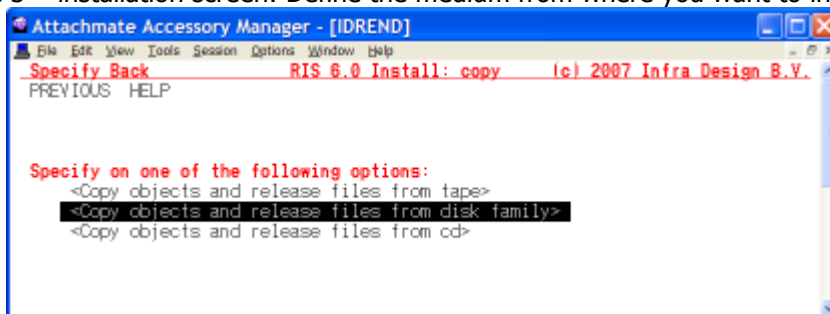
Please specify on the line 'Create a new repository' to choose that option.

The 2nd installation screen: Define the software that must be installed



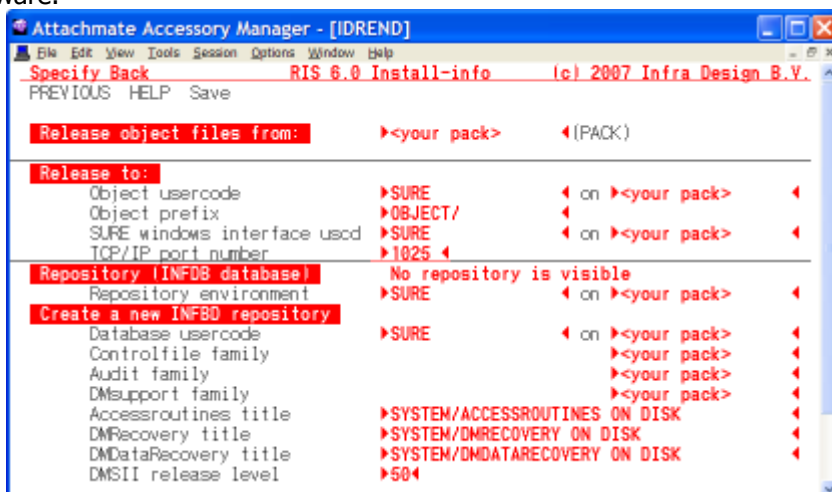
Please specify on the line 'Copy all objects' to choose that option.

The 3rd installation screen: Define the medium from where you want to install



Please specify on the line 'Copy from disk family'.

The 4th and last installation screen: Define the usercodes and packs where you want to place the software.



Your current usercode and packname are pre-entered on this screen.

Please read chapter of the installation.doc 9.2.7 to obtain detailed information about the installation screen.

- * Enter the fields on the installation screen to continue with the installation:

- **Location of the general SURE objects:**

Object usercode = '*' or SURE
 Object pack = The pack where you want to install the objects.
 Object prefix = The prefix of the objects (default = OBJECT/).
 In the example screen above the object location is '(SURE)OBJECT/= ON <your pack>'.

If you choose object-usercode = '*', then the SURE objects are copied to directory '*' and they will be visible for all usercodes who are working on this disk-family via the normal MCP visibility rules.

If you choose object-usercode = SURE, then the SURE objects are copied to that directory. If a SURE program has to be started from another usercode, then the name of the object has to be prefixed with usercode SURE (e.g. RUN (SURE)RIS/MENU).

For this demo installation a '*' must be entered in the object-usercode field.

- **Location of the objects of the SURE windows interface:**

| | | |
|-----------------------------|---|-----------------------------------|
| SURE windows interface uscd | = | SURE |
| SURE windows interface pack | = | Preferably your current diskpack. |
| TCP/IP port number | = | 1025 |

In the example screen above the object location of the SURE windows interface is `'(SURE)OBJECT/= ON <your pack>'`.

It is not necessary to install these objects in a global directory, because these objects are always started via a Coms program definition.

A TCP/IP port number must be entered to identify a port for the SURE windows interface. The port may not be in use by any other application. The default value is 1025. You are free to choose any other port number if you don't agree with the default value. Please notice that the port number also has to be defined at the SURE installation on the PC. Make sure that the same port number is used for the mainframe-installation and the PC-installation.

For this demo 'SURE' must be entered in field 'SURE windows interface uscd'.

- **Repository environment:**

The repository environment is defined as 'all usercodes that have access to repository-control-file RESPECT/TITLES via the normal MCP visibility rules'.

| | | |
|--------------------------|---|---|
| Repository env. usercode | = | '*' or SURE |
| Repository env. pack | = | The diskpack of installation usercode SURE. |

In the example screen above the repository environment is controlled by `'(SURE)RESPECT/TITLES ON <your pack>'`.

Notice that file RESPECT/TITLES must be visible for all usercodes that are going to access SURE. If only one usercode is going to access SURE, then this specific usercode can be entered in field 'repository environment usercode', otherwise an '*' must be entered which refers to the global directory '*'.

For this demo a '*' must be entered in the 'repository env. usercode' field.

- **Location of the database:**

| | | |
|-------------------------|---|--|
| INFDB database usercode | = | SURE. |
| INFDB database family | = | Any pack, not necessarily your current pack. |
| control-file family | = | Preferably equal to the INFDB database pack. |
| dmsupport family | = | Preferably equal to the INFDB database pack. |
| Audit file family | = | Choose a pack NOT equal to the INFDB pack. |
| DMSII release level | = | Your current DMSII release level. |
| Accessroutines | = | Enter your accessroutines program name |
| DMRecovery | = | Enter your Dmrecovery program name |
| DMDatarecovery | = | Enter your dmdatarecovery program name |

* Transmit the screen when you are satisfied about the options in the input fields.

At this moment the fields on the install screen should be entered like below:

Please notice the port number: in this example port 4764 is used, but you are free to choose any other port number

- * After you did the transmit, you must specify on button 'SAVE' (on the second line of the screen) to generate the installation job called WFL/RIS/INSTALL.
- * Start the installation job: START WFL/RIS/INSTALL
This job takes a few minutes to complete (depending on the power of your mainframe).

The installation job performs the following actions:

- Copy all objects and database files from the tape to the correct locations.
- Compile and install database INFDB.
- Authorize the installation usercode for all SURE commands and functions.
- Generate some jobs that you need during the demo.
- Give a message when the installation is completed.

The following files are now globally installed (in directory '*'):

- *RESPECT/TITLES on <your pack>
- *OBJECT/RIS/= on <your pack>
- *OBJECT/RESPECT/= on <your pack>

All other files are installed in the installation directory (SURE):

- (SURE)DASDL/INFDB
- (SURE)INFDB/=
- (SURE)OBJECT/RIS/API/=
- (SURE)DMSUPPORT/INFDB
- (SURE)DESCRIPTION/INFDB
- (SURE)RECONSTRUCT/INFDB
- (SURE)RISAPPLICATION/TITLES/WINDOWS



Please notice again the importance of repository-control-file RESPECT/TITLES:

RESPECT/TITLES is a seqdata file and you can modify or list the file via Cande.

The record that starts with key INFDB-LOCATION gives the location of the database control file.

The record that starts with key OBJECT-LOCATION gives the location of the SURE-objects.

When a program is started, the first thing it does is opening the visible respect/titles file to obtain the SURE-database location, and to obtain the SURE-object-location. The second step is to link to library RESPECT/LIBRARY in the SURE-object-location. The third step is to open the SURE-database. The SURE-object-location is also used when a SURE-object starts another SURE-object. This procedure makes it possible that SURE can be installed in a very flexible way in whatever any database-location and SURE object-location.

2.1.2 COMS Definitions

The SURE package has two interfaces to the user:

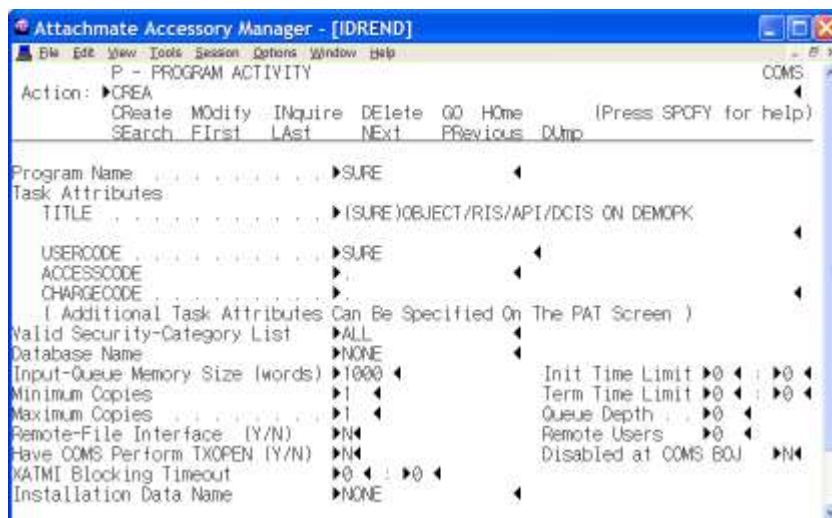
- A terminal emulation interface. On-line program OBJECT/RIS/MENU presents 24x80 screens on your terminal using a remote file.
Program OBJECT/RIS/MENU can be started via Cande (the Cande RUN command).
This interface is not used during this demo.
- A windows interface. A SURE PC application is installed on your PC, and communicates to the mainframe via a TCP/IP port.
A 'SURE windows interface' program has to be defined via COMS UTILITY, before SURE can be used on your PC.

- * Define the COMS utility attributes for the SURE-Windows interface and enable the COMS programs.
?ON UTILITY

The following program must be declared in COMS UTILITY:

A COMS PROGRAM definition for the SURE-windows-interface: program SURE

- You can choose any name as the program name in Coms, but it is easy to remember when the program name in Coms is equal to the usercode where the SURE-windows-interface is installed. And that is in this case SURE.

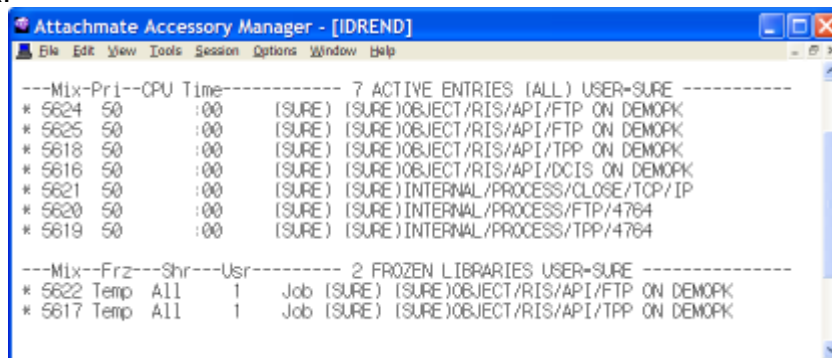


- Task attribute TITLE must be equal to the full name of program OBJECT/RIS/API/DCIS.
- Please notice that the program name in the TITLE contains a usercode and a pack, both equal to the usercode and pack of the SURE windows interface on the SURE-installation screen.
- Task attribute USERCODE must be: the installation usercode = SURE

The SURE windows interface must be started via a Coms-enable:
<Mixnr Coms>SM ENABLE PROGRAM SURE

The SURE windows interface can only be closed by a Coms-disable:
<Mixnr Coms>SM DISABLE PROGRAM SURE

- 🔔 After you enabled the COMS program then at least the following objects must be active in the mix:



```

Attachmate Accessory Manager - [IDREND]
File Edit View Tools Session Options Window Help

---Mix-Pri--CPU Time----- 7 ACTIVE ENTRIES (ALL) USER-SURE -----
* 5624 50 :00 (SURE) (SURE)OBJECT/RIS/API/FTP ON DEMOPK
* 5625 50 :00 (SURE) (SURE)OBJECT/RIS/API/FTP ON DEMOPK
* 5618 50 :00 (SURE) (SURE)OBJECT/RIS/API/TPP ON DEMOPK
* 5616 50 :00 (SURE) (SURE)OBJECT/RIS/API/DCIS ON DEMOPK
* 5621 50 :00 (SURE) (SURE)INTERNAL/PROCESS/CLOSE/TCP/IP
* 5620 50 :00 (SURE) (SURE)INTERNAL/PROCESS/FTP/4764
* 5619 50 :00 (SURE) (SURE)INTERNAL/PROCESS/TPP/4764

---Mix--Frz--Shr--Usr----- 2 FROZEN LIBRARIES USER-SURE -----
* 5622 Temp All 1 Job (SURE) (SURE)OBJECT/RIS/API/FTP ON DEMOPK
* 5617 Temp All 1 Job (SURE) (SURE)OBJECT/RIS/API/TPP ON DEMOPK
  
```

The mainframe installation of SURE is now complete.

2.1.3 Run the On-line Program (terminal emulation)

- * The SURE package can be started after the mainframe installation is completed correctly by running the on-line program:
RUN RIS/MENU

This program is only started to verify that the installation on the mainframe was successful. Please call ITSforSURE if problems arise at this moment.

The following messages appear on your screen:

1. Please wait, loading (SURE)INFDB ON <database pack>
2. This program runs for environment DEVELOP.

The first message only appears if the database was not yet active. On-line program OBJECT/RIS/MENU initiates library OBJECT/RESPECT/LIBRARY, and this library keeps the database up until the library is stopped via ODT command <mixnr>THAW.

The second message is given each time a program is started.

The terminal emulation on-line program presents the following menu screen:



The first line of the menu screen gives the following information:

- | | |
|-----------------------------|--|
| - <Spcfy back> | A specify on this place stops the program. |
| - SURE release number | This is the release indication of the software. |
| - User identification | The user who started the program. |
| - Repository identification | The name of the repository, indicated by the usercode of the database. |

- * RIS/MENU can be terminated as follows:
 - Keep specifying on <Spcfy back> until the program stops.
 - Press the 'home' button and enter 'GO END <xmit>' from the home position to stop immediately.

2.2 Installation of SURE on the PC

2.2.1 Copy and install the PC-software

Please follow this procedure to install SURE on a single (standalone) PC, and enable it to communicate with the SURE windows interface on the mainframe.

- * Prepare a PC with the following specifications:
 - Processor : 80486 or better
 - Memory: at least 8 Mb, preferably 16 Mb
 - Hard disk : at least 6Mb free space
 - Software : Windows 7 or newer
 - A TCP/IP connection to your mainframe
- * Start the installation as follows:
 - Switch to the directory where you unzipped the downloaded zip file
 - Double click on program INSTALL.EXE to start the installation procedure.

The 1st installation dialogue: kind of installation:



Please choose 'Install new application'.

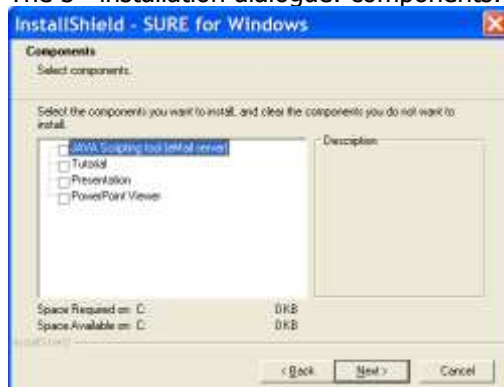
The 2nd installation dialogue: the path:



The default path is C:\SURE but you can choose any other path you like. We advise to install the client software in a sub-directory of C:\SURE. This sub-directory identifies the reason why this SURE client is installed. In that way it is easy to install multiple SURE clients for various reasons (for example a live installation and a test installation).

In this example the client software is installed in directory C:\SURE\DEMO\.

The 3rd installation dialogue: components:



Please select 'Tutorials' and 'Presentations' to install these components too.

The 4th installation dialogue: options:

This demo installation does not require any options.

The 5th installation dialogue: connections:

Please enter the IP address of your host followed by the TCP/IP port number in the format "nnn.nnn.nnn.nnn/pppp". If you are using a LMHOSTS file then you can enter the name of the host followed by the TCP/IP port number in the format "<hostname>/pppp".

In this example the connection is IDREND/4764.

- IDREND is the hostname in the LMHOST file on my PC.
- 4764 is the port number that is also used during the installation of the mainframe software.

The 6th and last installation dialogue: entry in the programs list.

This creates an entry in the program start menu on your PC. The default is 'SURE for Windows' but it is possible to create sub-directories like 'SURE for Windows\R&D' or 'SURE for Windows\P&O' when multiple repositories are available on the mainframe (a P&O repository and an R&D repository).

In this example the program name is SURE for Windows\DEMO.

Installation confirmation screen:



After these dialogues, a confirmation screen is presented, with a summary of the installation that will take place.

Press <Install> to start the installation-batch.

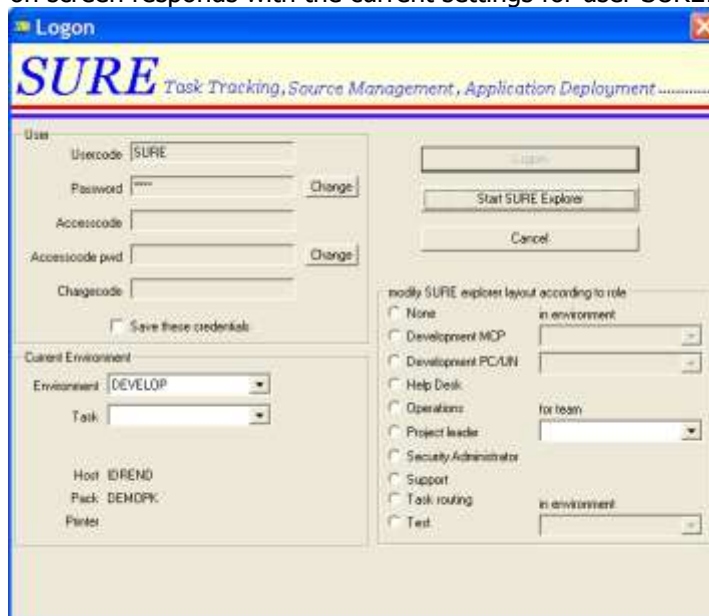
Wait until the installation-batch is finished.

The installation is now complete; you are ready to work with SURE.

2.2.2 Run SURE

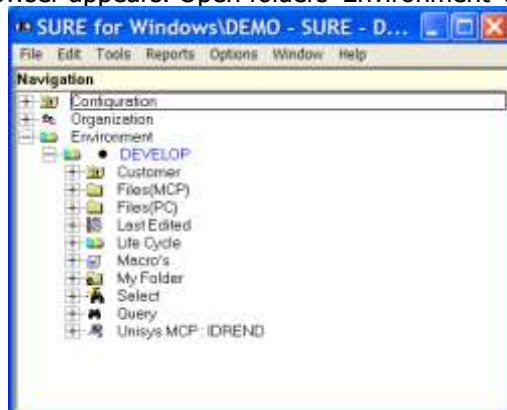
- * To start working with SURE
 - Go to the START-menu of your PC and open
All Programs → SURE for Windows → DEMO → SURE
 - Log-on with usercode SURE and the correct password.

The log-on screen responds with the current settings for user SURE:



For this moment we still have to log on with installation usercode (SURE) because that is at this moment the only known usercode in SURE. One of the first actions that we are going to do is to define other usercodes in SURE.

- * Click on button 'Start SURE explorer' to confirm the logon screen and to continue. The SureBrowser appears. Open folders 'Environment' and 'DEVELOP' to obtain the following window:



SURE shows all available environments as directory levels. Environment DEVELOP is opened, and details of this environment are given on the screen.

Help explains the keyboard and mouse support.

3 Options, Parameters and Securities

At this moment the first part of the installation is done: the software is copied and installed on the mainframe and on the PC, and we ran the software (on the mainframe and on the PC) to do a first check if it works correctly.

If you encountered any problems during this first test, then please contact us.

The second part of the installation is to set options, parameters and securities.

Multiple options and parameters are available where the user can change the behavior of SURE according to his own needs. Most of these options are of less importance for this demo or are pre-defined by ITSforSURE. Those options are not discussed during this demo.

Some other options have to be defined by the customer as part of the demonstration. This gives you automatically a global overview of the functionality of SURE, and you become familiar with the SURE explorer.

Options, parameters and securities have to be defined via the SURE-explorer. These options and securities are used by the SURE-explorer as well as the mainframe programs (on-line and batch).

3.1 Define an employee-function

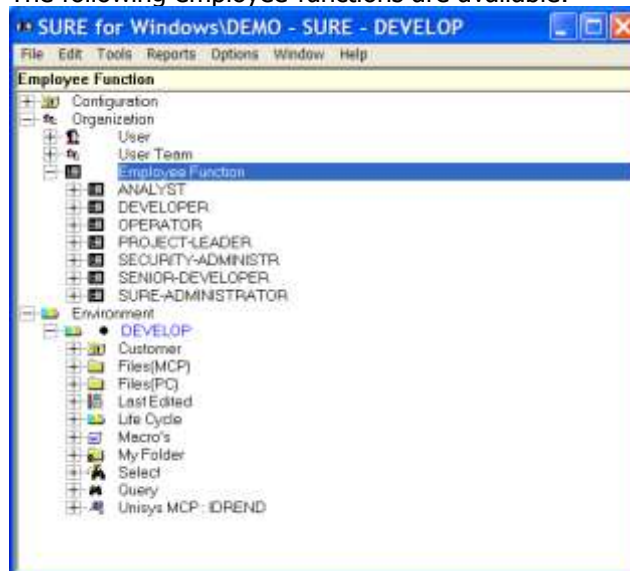
It is possible to define employee-functions in SURE. Employee-functions are linked to usercodes. An employee-function has two purposes:

- Authorization of SURE functions, for example: Users with this employee-function are allowed to perform these functions.
- Routing, for example: this task has to be solved by a user with this employee-function.

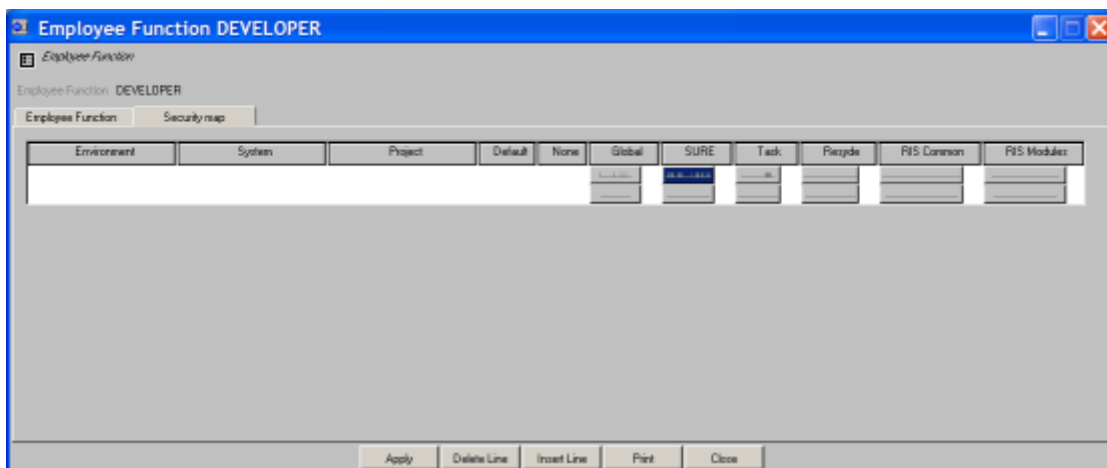
During the installation some employee-functions are automatically defined.

★ **The next step is to get familiar with employee functions:**

- Open folder 'Organization' → Employee-function
- The following employee-functions are available:



- Click on the icon of DEVELOPER:  DEVELOPER , to open the properties of that employee-function.



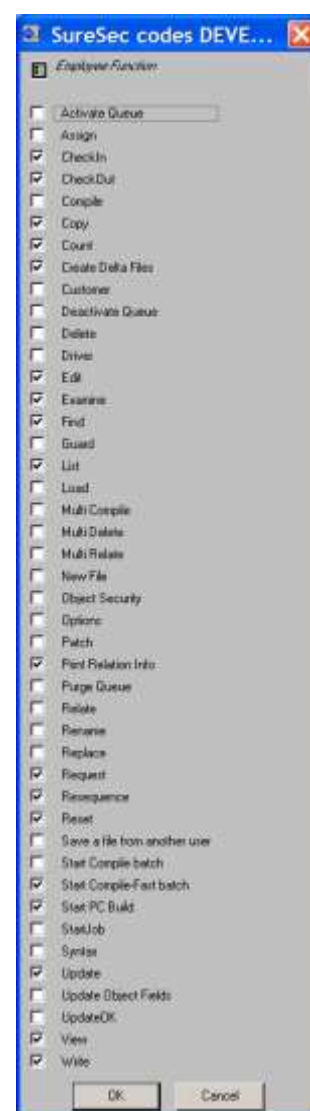
- Click on tab <Security map>

It is possible to define security maps per environment. In this example there is no environment name entered in the 'Environment' column, and in that case the security map is global and applicable for all environments.

- A click on the button in column SURE shows the current settings of SURE authorizations, for employee function DEVELOPER. The screen on the right hand side is an example of this. The bitmap on the button reflects the options 'inside' that button that are enabled.

The entire security-map is the combination of all six buttons on the row. During this demonstration only the buttons <Global>, <SURE> and <Task> are relevant.

- You can modify the security map to your needs. For example: function 'COMPILE' is a valid function for a developer, so you can enable that function.
- Click on button <Apply> do send your modifications to the repository.
- Click on button <Close> to leave the properties screen.




*** The next step is to define a new employee function TASK-MANAGER:**

- Right-click on the word 'Employee-function' and choose <New> from the menu:



- Add employee-function TASK-MANAGER
- Click on button <Apply> to add the employee-function
- Click on and button <Close> to close the dialogue.

3.2 Define a personal usercode with authorizations

 During the installation phase, the initialization usercode (SURE) is added in the repository with ALL authorizations. The only reason why this was done was to unlock the system for the first usercode. Other usercodes do not have any authorizations at this moment and can not do any update activities in the system. Using the initialization usercode, it is now possible to define other usercodes with authorizations.

Please read chapter 35 of the manual to obtain detailed information about the authorization mechanism.

The best way to define authorizations is via employee-functions (or security categories): authorization maps are linked to employee-functions, and these employee-functions are linked to usercodes. Consequently the usercode inherits all authorizations that are defined for its employee-functions.

Each individual person that works with SURE must have his own personal usercode. This is necessary for authentication purposes, logging purposes, task-routing purposes, et cetera.

During this demo the Cande work environment is defined by usercode (DEV). This means that all files-in-maintenance are placed on disk under usercode (DEV). A programmer logs on to SURE with his personal usercode, but he accesses the files in Cande-directory (DEV).

Each personal usercode must also be known in the userdatafile.

It is possible to use usercode (SURE) as a personal usercode, but we don't advise this. If multiple persons are going to work with the SURE system, then each person must have his own personal usercode, otherwise the actions of one person may jeopardize actions of another person. A disadvantage of using (SURE) as your personal usercode is that the installation environment of SURE is going to be mixed with your application work environment. Files of two totally different systems will appear in the same Cande directory. It won't give any problems as long as you don't overwrite any of the SURE installation files, but it can be confusing.

*** The next step is to define a personal usercode in SURE:**

- Log on with the installation usercode (SURE).
- Open folder Organization
- Right click on folder <User> and choose <new> to create a new personal usercode. This usercode must be known in the userdata file. The following dialogue is given where specific usercode attributes can be entered:

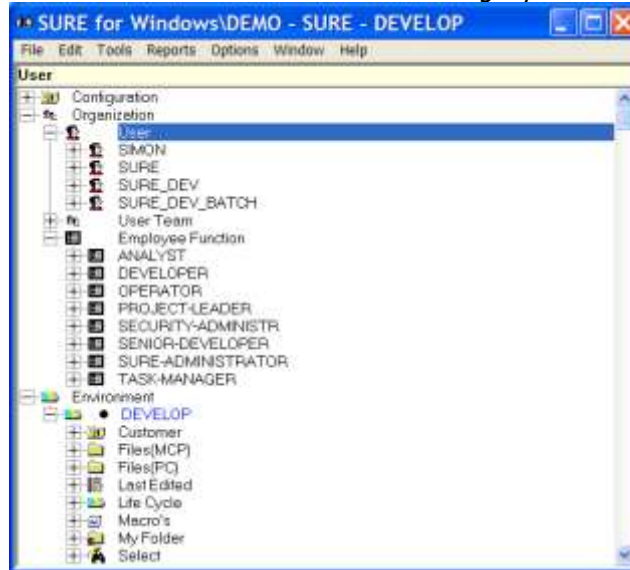
| | |
|---------------------|---|
| User Name | The usercode (must be known in the user data file). |
| Name | The normal name of the person (e.g. Simon Groot) |
| Email address | Empty: not used during this demo |
| Department | Empty: not used during this demo |
| Initials | The initials of the person (e.g. SG) |
| Default environment | DEVELOP |
| Printer | Empty: not used for the moment |
| Default language | English |
| Employee function | Empty: not used for the moment |
| Team Name | Empty: not used for the moment |

- Tab sheet 'Security map' shows the authorization bits that are set for this user. Do NOT set any authorization bits for this user, because they must be inherited from employee-functions.
- Click on button <Apply> to add the new usercode.
- Click on button <Close> to leave the screen.

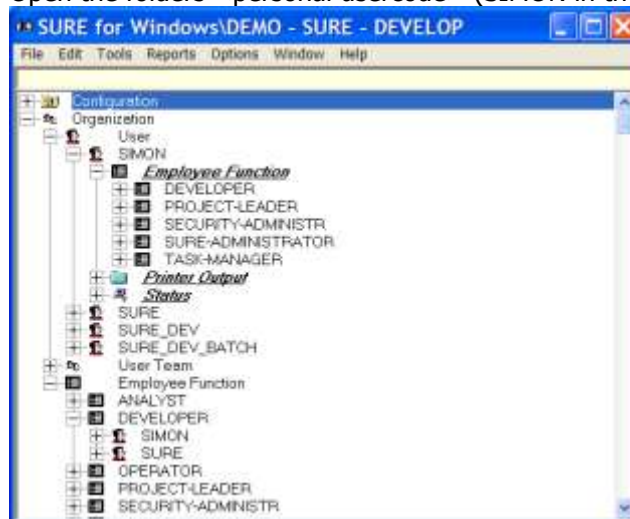
*** The next step is to link your usercode to some employee-functions:**

- Open folders 'User' and 'Employee-function'
- Refresh the user list as follows:
 - o Click on one of the usercodes and press F5
- Refresh the list with employee-functions the same way: Click on an employee-function and press F5.

- The SURE browser has now the following layout:




- Drag your personal usercode (SIMON in the example screen) with your mouse and drop it on employee-function DEVELOPER. You are now a developer.
- Drag/drop your personal usercode also to employee-function SECURITY-ADMINISTR. From now on we don't need to log on anymore with usercode (SURE) because we can now maintain usercodes and securities with our personal usercode.
- Link your personal usercode also to employee-functions PROJECT-LEADER, TASK-MANAGER and SURE-ADMINISTRATOR. You now have full authorizations.
- Open the folders <personal usercode> (SIMON in the example) and DEVELOPER:

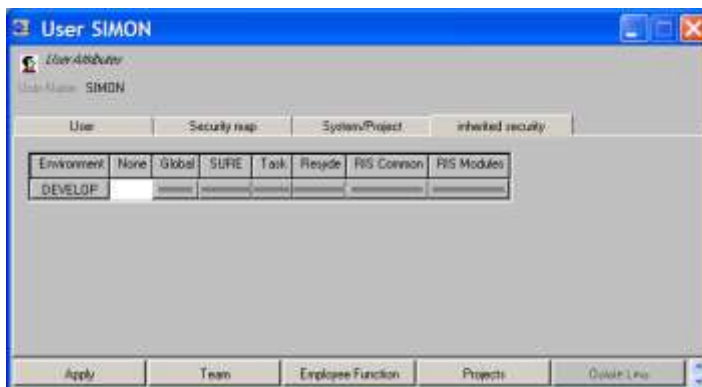


- User SIMON has employee-functions DEVELOPER, PROJECT-LEADER, SECURITY-ADMINISTR, SURE-ADMINISTRATOR and TASK-MANAGER. Employee function DEVELOPER has user SIMON (and SURE).

*** The next step is to check the authorizations of your personal usercode:**

Your usercode inherits now all (the combination) the securities that are defined for his employee functions.

- Click on the icon of your usercode:  SIMON
- Click on tab <Inherited security>



-
- You can already see via the bitmap on the buttons that all the securities are enabled, but when you are not sure you can click on one of the buttons.
- Click on button <Close> to leave the screen.

*** It is now time to log on to SURE with your personal usercode.**

- Click on <SURE menu bar> → File → Log off
- Log on to SURE with your newly defined personal usercode.

If your personal usercode is not equal to the installation usercode (SURE), then that installation usercode is not longer used during this demo. In general the only activities that are done under the installation usercode (SURE) are:

- Compilation of the INFDB dasdl, and other DMUTILITY work for database INFDB.
- Installation of new releases.

3.3 Environments

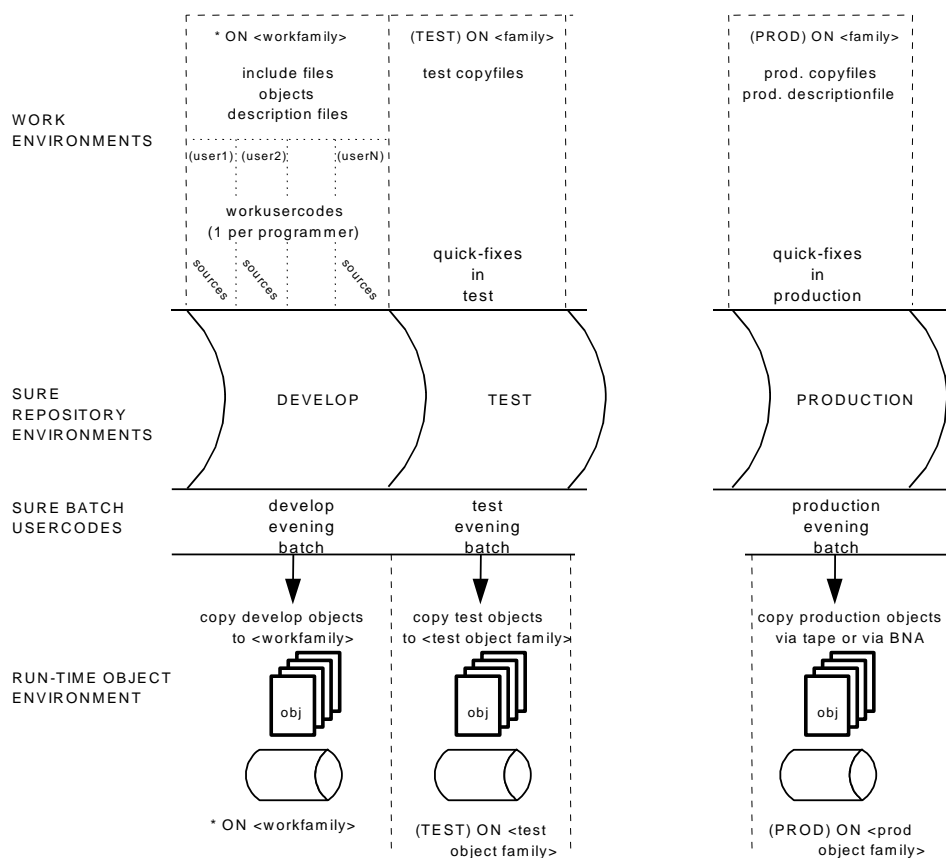
In this section we are going to define extra environments in the repository, and (optionally) we are going to rename environments.

- * Log on to SURE with your personal usercode.

The demo repository contains now only one environment: DEVELOP.
A Cande work-location is linked to SURE environment DEVELOP: this is usercode DEV.
The SURE evening batch for DEVELOP has to run under usercode DEVBATCH.

SURE can be installed in various ways and has its impact on the work-locations of programmers, on the directory where common and visible copy-files are placed, on the run-time directory where objects are installed, and on compilation and object-transfer (deploy) procedures.

The following picture shows a possible installation of SURE.




- Notice the work locations of programmers (upper left hand corner): each programmer has his own individual usercode, and copy-files are placed in directory '*'.
- Notice the run-time object environment for DEVELOP (lower left hand corner): the objects are placed under directory '*' on the work pack of the programmers.
- Notice the run-time object environment of PRODUCTION: this environment can be on another host. Objects are transferred via library maintenance (tape or BNA).
- Notice the work environment of quick-fixes for test or production: only one usercode directory can be enough for this environment.

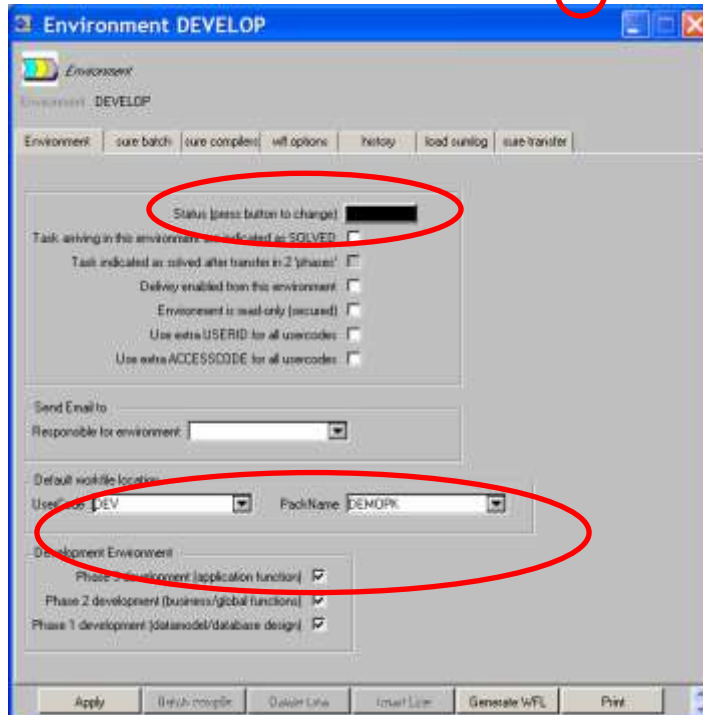
Summarized: Each SURE environment contains a complete set of application programs, and must be linked with a Cande environment where the source maintenance must be done (work location), and a directory where the application objects are placed (run-time location).

When a package is installed for demonstration purposes only, then most sites want to have the files installed in as less directories as possible (and certainly not in a global directory '*'), because then it is easier to uninstall the package at a later stage.

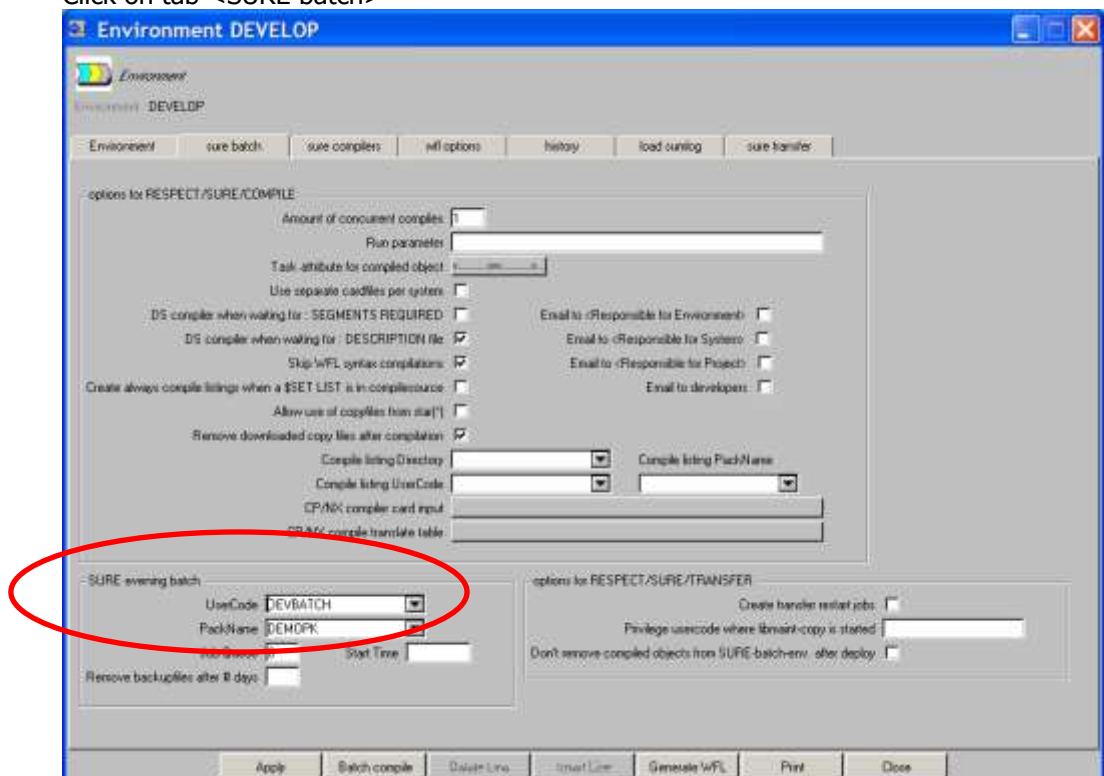
3.3.1 Verify the definition of environment DEVELOP

★ **The next step is to verify the properties of environment DEVELOP:**

- Click on the icon of environment DEVELOP:  **DEVELOP**



- Change the color of this environment to red (via button Status)
- Change the default workfile location to: usercode DEV and pack <your pack>.
- Click on tab <SURE batch>



- Change the SURE evening batch location to: usercode DEVBATCH pack <your pack>
- Click on button <Apply> to change the environment properties.

The locations of SURE environment DEVELOP are now correctly defined:

- The default work-location is (DEV) ON <your pack>.
- The location where the SURE batch runs is (DEVBATCH) on <your pack>.

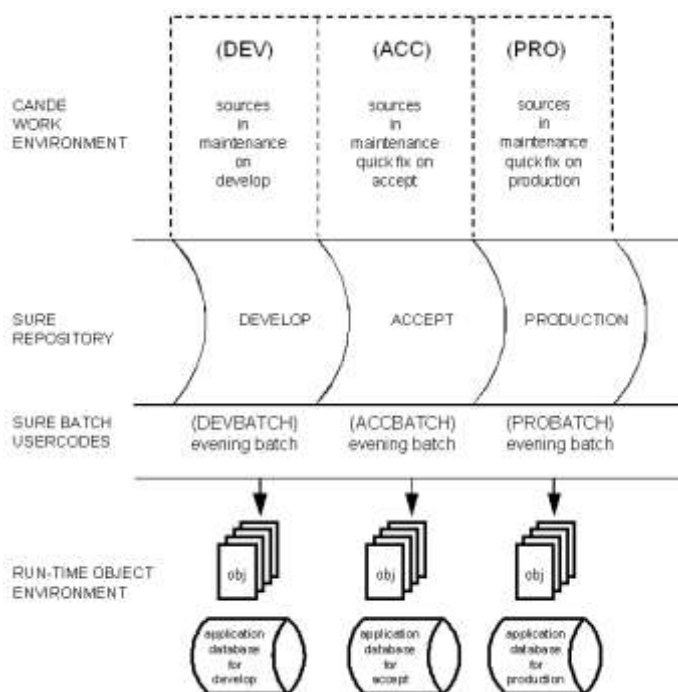
Note: if you are planning to use this environment as a test environment, we advise you to use the original default workfile location and SURE evening batch location. By doing this the security administrator does not have to manually create these user codes.

3.3.2 Define two new environments: ACCEPTANCE and PRODUCTION

We are going to add two extra environments in SURE:

- Environment ACCEPTANCE with work-location ACC and batch-location ACCBATCH.
- Environment PRODUCTION with work-location PRO and batch-location PROBATCH.

The following picture represents this situation:



*** The next step is to create two extra environments. First ACCEPTANCE:**

- Right-click on environment-name DEVELOP and choose 'New' from the menu.
- This opens the add-environment dialogue where the new environment is added as a clone of DEVELOP:

- Enter/change the following fields to define the new environment correctly:
Tab sheet 'Environment':
 - Environment = ACCEPTANCE
 - Environment BELOW newly created environment = DEVELOP
 - Click on radio-button 'copy upwards'
 - Click on button 'Status' to define a color color yellow
 - Default workfile location usercode = ACC
- Tab sheet 'SURE batch':
 - SURE evening batch usercode = ACCBATCH

- Click on button 'Apply' to start the creation of the new environment and click on the OK button of the verification dialogue:

- The new environment ACCEPTANCE is now created by a batch process on the mainframe. This may take a few moments.
- The new environment is not yet known in your SURE browser. Therefore you have to log off and log on again.
- Click on Menu bar → File → Logon.
- Environment ACCEPTANCE appears now in the SURE browser.
- When the create-environment batch is finished, the newly created environment is in a read-only state. This read-only bit must be switched off to make the environment available for the programmers.
- Right-click on environment name 'ACCEPTANCE' and choose 'Properties'.
- Reset the option <Environment is read-only (secured)> on tab-sheet <Environment>.
- Click on button <Apply> to change the environment properties
- Click on button <Close> to leave the screen.

★ **Repeat the last actions to create environment 'PRODUCTION':**

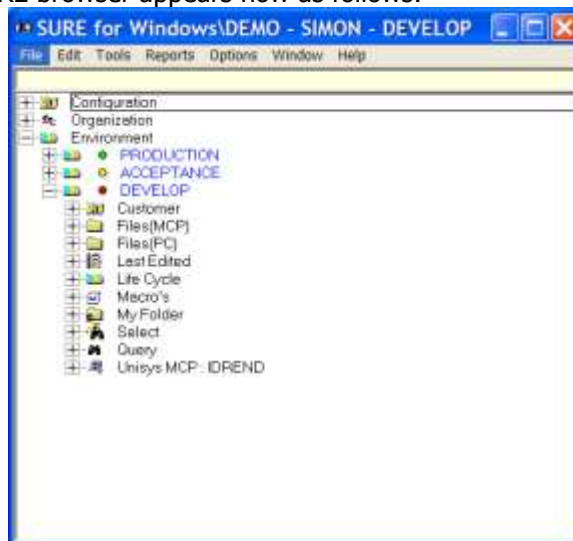
- In this case environment ACCEPTANCE must be copied upwards as PRODUCTION.
- Right-click on environment-name 'ACCEPTANCE' and choose 'New' from the menu.:
- Enter/change the following fields to define the new environment correctly
 - Tab sheet 'Environment'
 - Environment = PRODUCTION
 - Environment BELOW newly created environment = ACCEPTANCE
 - Click on radio-button 'copy upwards'
 - Default work file location usercode = PRO
 - Click on button 'Status' to define a color color green
 - Tab sheet 'SURE batch'
 - SURE evening batch usercode = PROBATCH
- Click on button 'OK' to start the creation of the new environment and click on the OK button of the verification dialogue.
- Click on Menu bar → File → Logon to refresh. Environment PRODUCTION appears now in the SURE browser.
- Right-click on environment name 'PRODUCTION' and choose 'Properties'. Reset option <Environment is read-only Secured>.
- Click on button <Apply> to change the environment properties.
- Click on button <Close> to leave the screen.

All environments that are necessary for the demo are now defined in SURE.

Read chapter 37.2.2 of the SURE manual for detailed information about work environments.

Read chapter 37 to obtain a global overview about environments and security aspects.

The SURE browser appears now as follows:



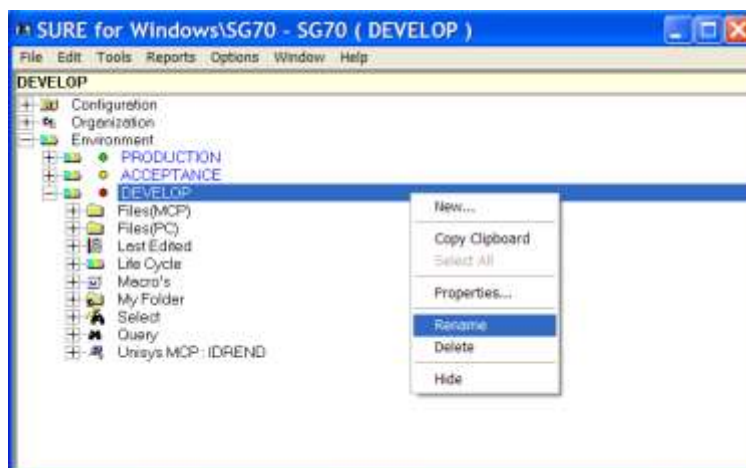
- Notice the environments DEVELOP, ACCEPTANCE and PRODUCTION.
- Notice the colors.

3.3.3 Rename environments to your own needs

It is possible to rename environments to your own standards.

★ **Rename an existing environment (example)**

- Rightclick on the environment-name that you want to change, and choose "Rename" from the menu:



- The following rename screen is given:



- In this example environment DEVELOP is renamed as DEVELOPMENT, but you can use any name you want.
- Click on OK to start the rename-action.
- A verification is required for the rename-action:



- Click on button OK
- Close and restart the SURE browser via File → Logon
- The environment is renamed:

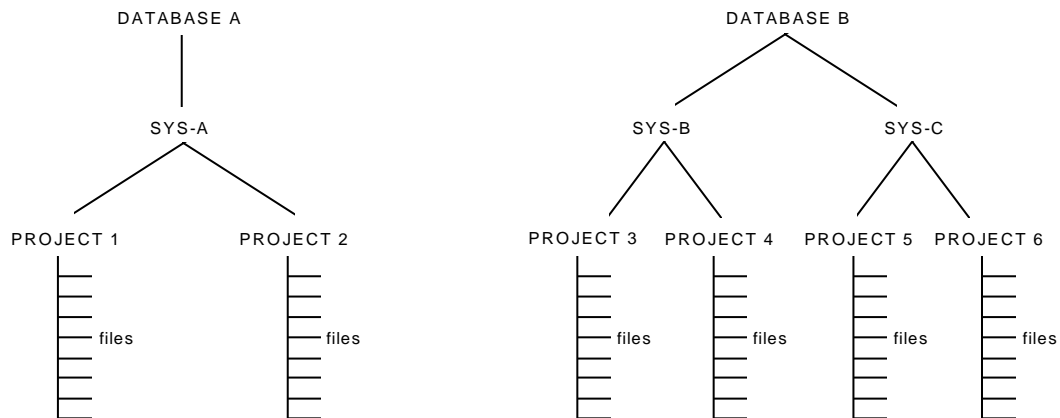


In this demo we will continue with the **old** environment name: DEVELOP

3.4 Define an application system

The total set of application programs stored in SURE is usually divided in several subsets. We call such a subset a **SYSTEM**, and all programs that belong to the same subset are linked to the same system. A program cannot be linked to multiple systems at the same time. The user can decide which programs belong to which system.

Application systems are often subdivided in several projects (or sub-systems). A project cannot be a sub project of multiple systems. The following picture shows this situation:

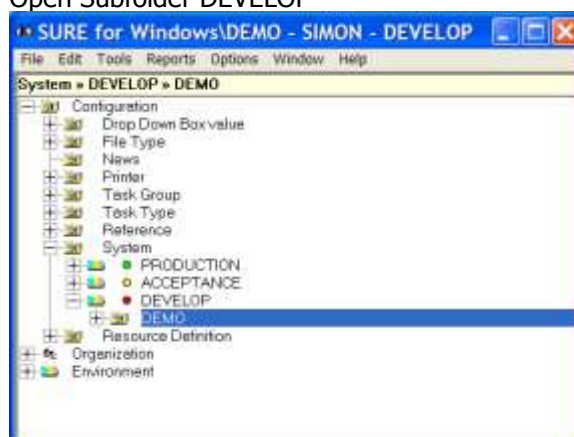


Cande work-locations can be defined per application system and per SURE environment. This makes it possible to develop different application systems on different packs.

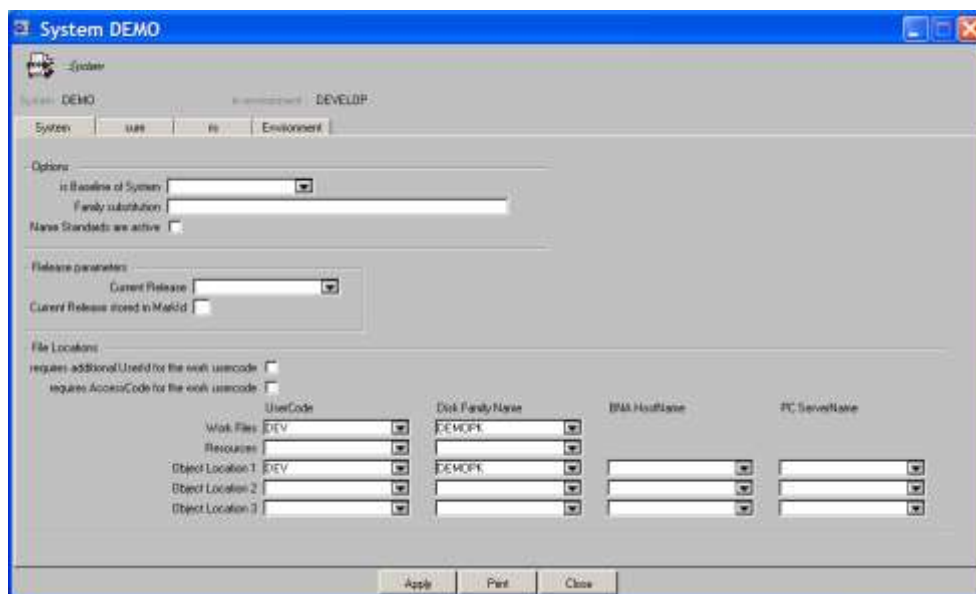
At this moment the repository contains only system SYS1. It is now time to define application system DEMO.

*** The next step is to define system DEMO:**

- Open folder Configuration
- Right click on subfolder <System> and choose <New>.
- Enter DEMO as a new system name in the pop-up window and click on <OK>.
- Open now folder <System>
- Open Subfolder DEVELOP



- Click on the icon of system DEMO to open its properties:  DEMO



- Enter/change the following fields in environment DEVELOP
 - Tab sheet 'System'
 - Work files usercode = DEV This creates a specific work-location for system DEMO
 - Work files pack = <your pack>
 - Object files usercode = DEV This creates a specific object-location for system DEMO
 - Object files pack = <your pack>
 - Tab sheet 'Sure'
 - Set option 'Put copy files in work directory' Changed copy-files are now put in the work-location (DEV) ON DEMOPK
- Click on button <Apply> and then on button <Close> to leave the window.

The work-location and object-location of system DEV are now defined for the DEVELOP environment. The next step is to define them for the ACCEPTANCE and PRODUCTION environments.

*** Define the properties for system DEMO in ACCEPTANCE:**

- Open subfolder <ACCEPTANCE>
- Right-click on system DEMO (in environment ACCEPTANCE) and choose 'Properties'
- Enter/change the following fields to define system DEMO in environment ACCEPTANCE:
 - Tab sheet 'System'
 - Work files usercode = ACC
 - Work files pack = <your pack>
 - Object files usercode = ACC
 - Object files pack = <your pack>
 - Tab sheet 'Sure'
 - Set option 'Integrity mechanism is active'

*** Define the properties for system DEMO in PRODUCTION:**

- Open subfolder <PRODUCTION>
- Right-click on system DEMO (in environment PRODUCTION) and choose 'Properties'
- Enter/change the following fields to define system DEMO in environment PRODUCTION:
 - Tab sheet 'System'
 - Work files usercode = PRO
 - Work files pack = <your pack>
 - Object files usercode = PRO
 - Object files pack = <your pack>
 - Tab sheet 'Sure'

- Set option 'Integrity mechanism is active'

3.5 Various task options

'Task' is an important entity in SURE. Without a task it is not possible to make modifications to the application software.

A task describes 'an error in the current application' or 'a request of a new feature'.

A number of options need to be defined before it is possible to add a task

Before we can use tasks we must define some task properties.

3.5.1 Priorities

- * Open folder Configuration → Drop-down-box-value.
- * Right-click on the word 'Task Priorities' and choose 'New' from the menu.
- * Add the following priorities:
HIGH
LOW

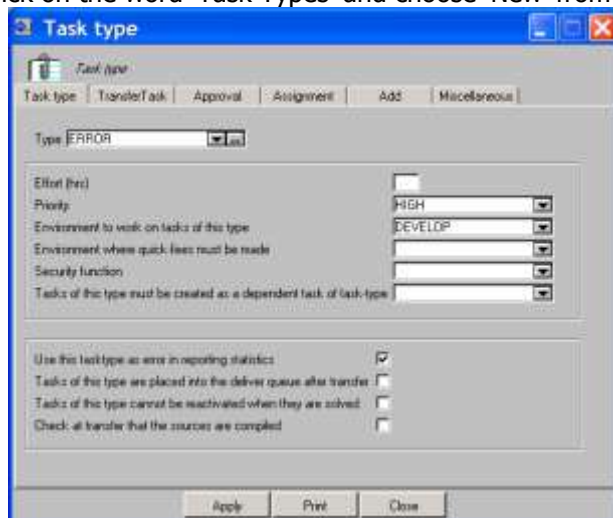
3.5.2 Task Types

Please read chapter 38.4 of the SURE manual to obtain a first idea about task types.

A task-type identifies the workflow of a task. The workflow of an emergency-fix differs from the workflow of a regular change-request.

Task-types are mandatory. It is not possible to define a task without a task-type.

- * Open folder 'Configuration'.
- * Right-click on the word 'Task Types' and choose 'New' from the menu.



- * Add the following task-types:

| | |
|-----------------------|---|
| Task Type = ERROR | |
| - Priority | = HIGH |
| - Environment | = DEVELOP |
| - Set option | ' Use as error in reporting statistics' |
| Task Type = FEATURE | |
| - Priority | = LOW |
| - Environment | = DEVELOP |
| Task Type = QUICK-FIX | |
| - Priority | = HIGH |
| - Environment | = PRODUCTION |
| - QuickFix level | = PRODUCTION |

- Set option 'Use as error in reporting statistics'

3.5.3 Task Groups

The purposes of task groups are:

- All tasks with the same group get a similar name
- A task inherits attributes from its group, so less fields to enter when defining a new task.

Task groups are optional.

- * Click on folder 'Configuration'
- * Right-click on the word 'Task Group' and choose 'New' from the menu.

- * Add the following task-groups:

| | | |
|---------------------------|---|-----------|
| Task Group = DEMO-ERROR | | |
| - Project | = | DEMO |
| - Type | = | ERROR |
| Task Group = DEMO-FEATURE | | |
| - Project | = | DEMO |
| - Type | = | FEATURE |
| Task Group = DEMO-QF | | |
| - Project | = | DEMO |
| - Type | = | QUICK-FIX |

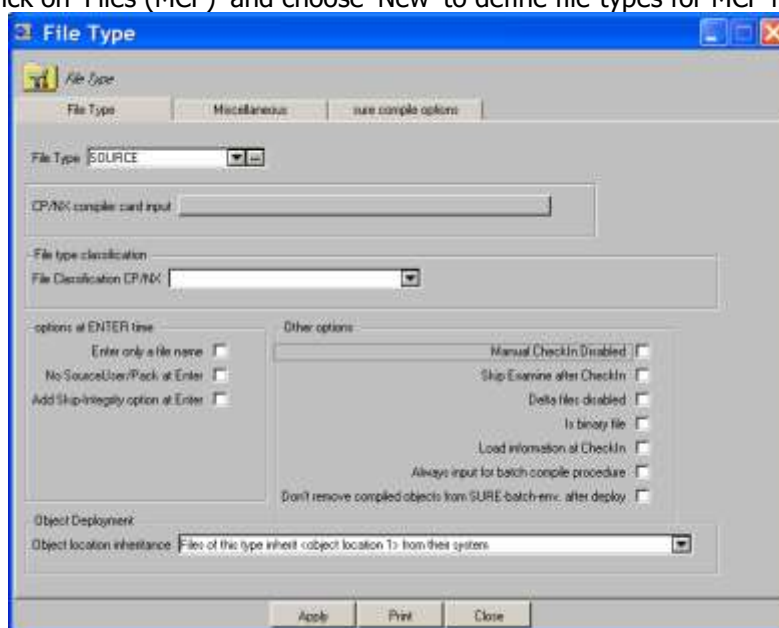
3.6 File Types

A FILE-TYPE defines a logical name for a type of file. This type is not related to the FILEKIND attribute of files on the Unisys CP/NX. The file-type determines how a file is handled by the SURE software.

A file can only be loaded in SURE if it has a valid file-type.

Please read chapter 34 of the SURE manual to obtain a first idea about file-types.

- * Open folder 'Configuration → FileType'
- * Right-click on 'Files (MCP)' and choose 'New' to define file-types for MCP files:



- * Add the following file-types for MCP files:
 - File Type = SOURCE
 - Object location inheritance Files of this type inherit <object location 1> from their system
 - File Type = INCLUDE
 - File classification Use as INCLUDE file
 - Object location inheritance Files of this type do not have an object location.
- * Right-click on 'Files (PC)' and choose 'New' to define file-types for PC files:
- * Add the following file-types for PC files:
 - File Type = BINARY
 - Set option 'Skip examine after check-in'
 - Set option Delta files disabled
 - Set option Is binary file
 - Object location inheritance Files of this type do not have an object location
 - File Type = TEXT
 - Object location inheritance Files of this type inherit <object location 1> from their system

3.7 File References

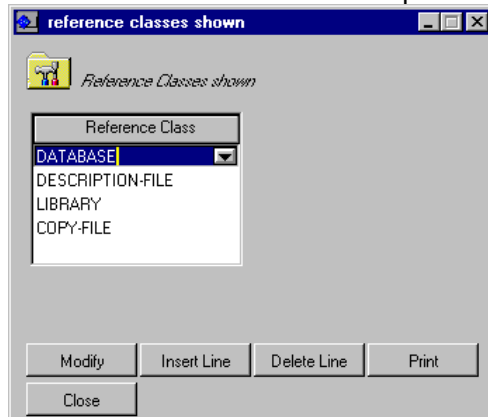
All files that are loaded in SURE are scanned for language statements that refer to other files (include-files, libraries, database, datasets, et cetera).

We call this scanning-process 'the examine function'

Please refer to chapter 26 of the SURE manual for detailed information about the examine function.

All interrelationships between sources files are stored in SURE. Via the 'Reference folder' it is possible to make a selection of these interrelationships (or program references), and to visualize the selected references via the SURE browser.

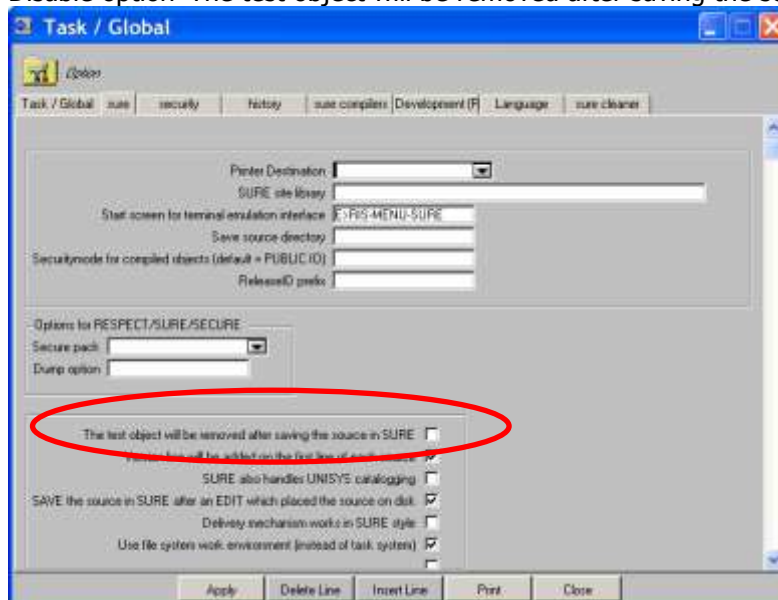
- * Open folder 'Configuration'
- * Right-click on 'Reference' and choose 'Properties' to select file-references for the SURE browser:



In this example, the following references are added: DATABASE, DESCRIPTION-FILE, LIBRARY and COPY-FILE.

3.8 Other repository options

- * For this demo we have to change a global option as well.
 - Right click on folder 'Environment' → Global options'
 - Switch to tab SURE
 - Disable option 'The test object will be removed after saving the source in SURE'



3.9 Local SURE options

Some options can be defined per work-station. This can be done via button 'Options' in the menu-bar.

The most important options are:

'SURE Options'/'Local Options':

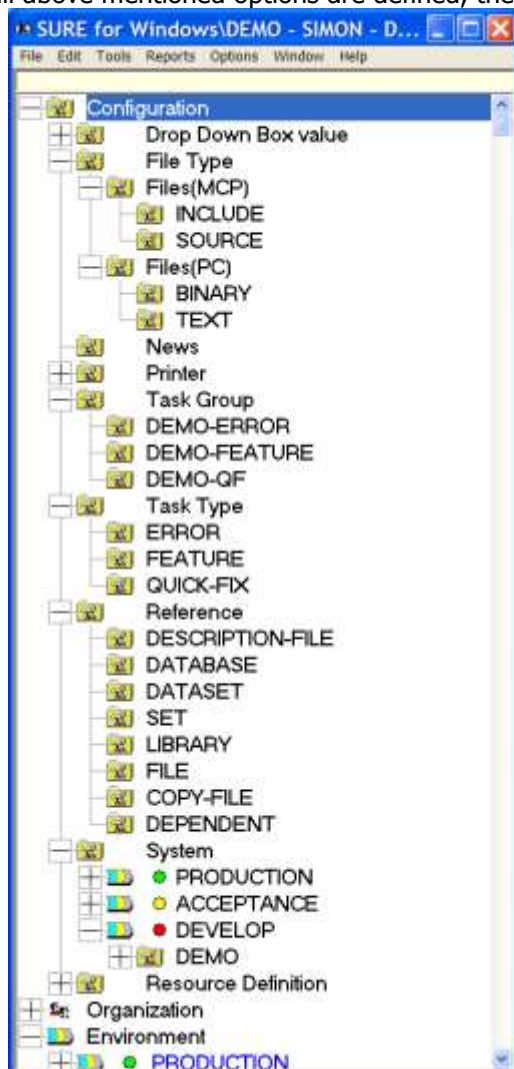
Utility for Compare: the default is WINMERGE.EXE (a shareware product).

Utility for File Editing: the default is NOTEPAD.EXE. We made interfaces to several other editors such as 'ED for Windows', MultiEdit, etc. Please read chapter 12.2 of the SURE manual for detailed information about the various editors that we support.

If your site uses another local PC editor, then please contact us.

3.10 Options verification

When all above mentioned options are defined, the SURE browser appears as follows:



The basic configuration of SURE is now done.

4 Task registration

Please read chapters 38.1 through 38.4 to obtain a first idea about tasks.

Each adaptation to software is linked to a task. Therefore a task must be added and made current before a programmer can start working.

4.1 Task Registration via SURE

- * Logon with your personal usercode.
- * Click on Edit (in the menu bar)
- * Click on New/Task. The following task definition screen appears:

This task-function is a small Visual-Basic application (plug-in) that can easily be customized to a sites specific need. For example if a site wants to keep extra task-info in the SURE repository that needs to be used in the SURE query language.

- * Add a new task:
A task is defined by entering a project and a task-type. The name of the task is determined by SURE
 - Project = DEMO
 - Task-type = FEATURE
 - Description = The first demo task

The newly entered task is called DEMO0001.

Notice the status of the newly entered task: ENTERED.

Notice the attribute 'Environment' (under button 'Extended data'): The environment is inherited from the task type.

Read chapter 36.5 through 36.6 of the SURE manual for detailed information about the fields on the task definition screen.

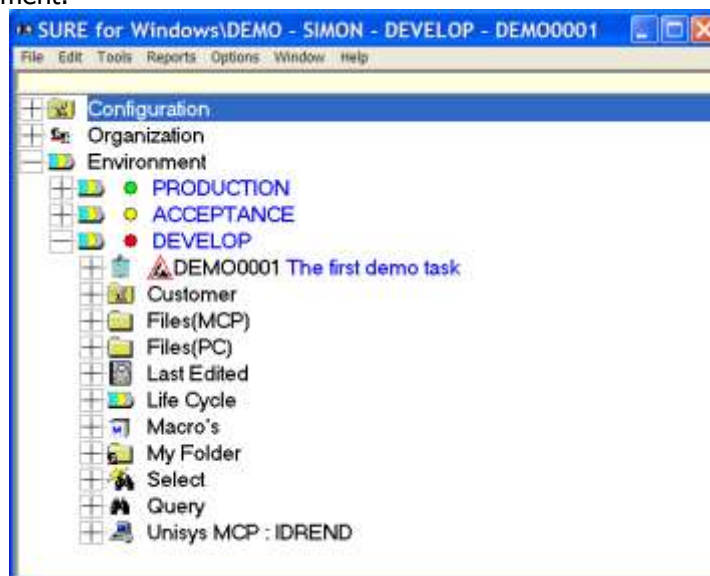
- * We want to start working on this task. This is only possible if the task is assigned to me. There are several ways to do this:

1. Click radio button 'I start working on this task' on the task definition screen. In this case the task becomes known as 'My current work task'.
2. Click radio button <Usercode>, <Function>, or <Team> and enter my usercode or one of my employee-functions or one of my team in the corresponding field (all in box <To Handle By>. The task is now assigned to me. It appears in my ToDo list but it is not my current work task.
3. The last added task is visible as a folder on the second line in the environment (DEVELOP). Right click on the task and choose option <Current>. In this case the task becomes known as 'My current work task'
4. Open folder LifeCycle → Tasks to Assign, and click with the right mouse button on the task that you want to assign. Choose <Assign> from the speed-menu followed by <Current> from the speed menu. In this case the task becomes known as 'My current work task'.
5. Open folder LifeCycle → Tasks to Assign, and click with the right mouse button on the task that you want to select. Choose <Assign> from the speed-menu. The task is now assigned to me, but it is not 'My current work task'

Tasks that are assigned to me appear in my ToDo list (folder Life Cycle → Tasks To Do).

The task that is marked as 'my current work task' is also assigned to me, but it appears also as the first folder in my environment, and in the SURE title bar. And it is as the default task pre-filled in the check-out screen when you start working on files.

The current task is displayed in the title bar of the SURE explorer, and as a folder in the DEVELOP environment:



- * Right-click on task DEMO0001 and choose 'log' from the speed-menu. The following screen appears:



- ★ Click on button 'Close' and double click on DEMO0001. The following screen appears:

- Notice the status of the task: DEVELOP/ASSIGNED (at the top of the screen).
- Notice that info-block 'To Handle By' contains now your usercode. This means that this usercode is the only person who can work on this task. It is also possible to enter an employee function in this field. In that case all users who are linked to that employee function can work on the task.
- Notice that fields 'Project' and 'Type' are grey. It is only possible to modify these fields if the task has status ENTERED or SOLVED.

- ★ Click on button 'Modify'/'Properties'
Change 'To Handle By' to: Function DEVELOPER

You are now ready to work on files!


5 Load a directory with files in SURE

The best way to understand the functionality of a software management package like SURE is when your own files are loaded in it.

We now ask you to load some files of your own application system in SURE. You are free to choose the files that you want to load, but for the demonstration it is important the loaded files meet the following conditions:

1. A small directory (5 to 10) with sources that must be compiled (= that have an object).
2. All copy-files (or include files) that are referenced by those programs.

It is beneficial of the demonstration if a database and/or library is referenced by your programs.

 Please ensure yourself that the loaded files form together a complete sub-system of your applications: all referenced copy-files are loaded in SURE.
Please do not load too many files for this demonstration.

- * Copy the files that you are going to load to the work environment: usercode DEV on <your-pack>.

The load procedure is explained via an example.

The following two directories with files are placed under work-usercode (DEV):

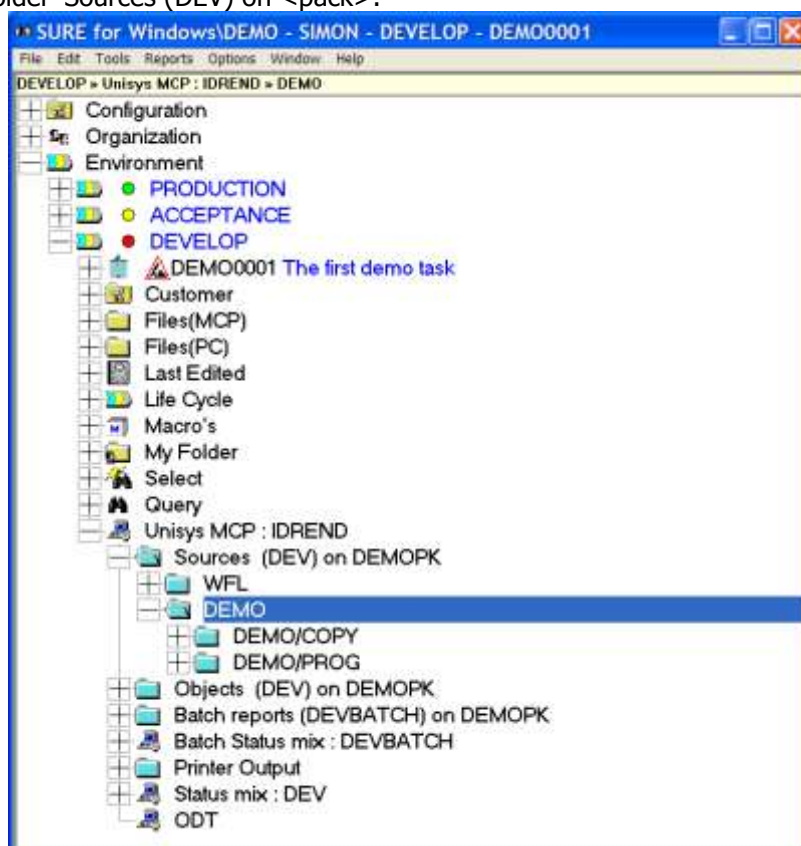
- DEMO/PROG/= a directory with program files.
- DEMO/COPY/= a directory with include files.

In the remainder of this paragraph you must substitute the directories named:

DEMO/COPY/= by <your directory with include files>

DEMO/PROG/= by <your directory with program files>.

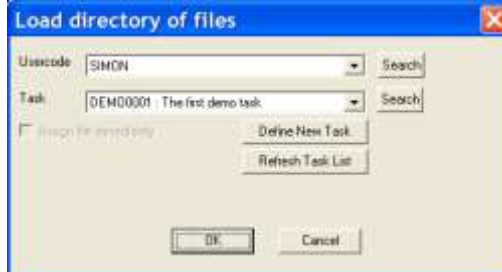
- * Open folder 'Unisys MCP <hostname>'
- * Open folder 'Sources (DEV) on <pack>':



All directories under usercode (DEV) are now given. You can click on one of the directories to go deeper in the directory-tree. In this example you see two sub-directories DEMO/COPY/= and DEMO/PROG/=.

Directory DEMO/COPY/= is the first directory that is going to be loaded.

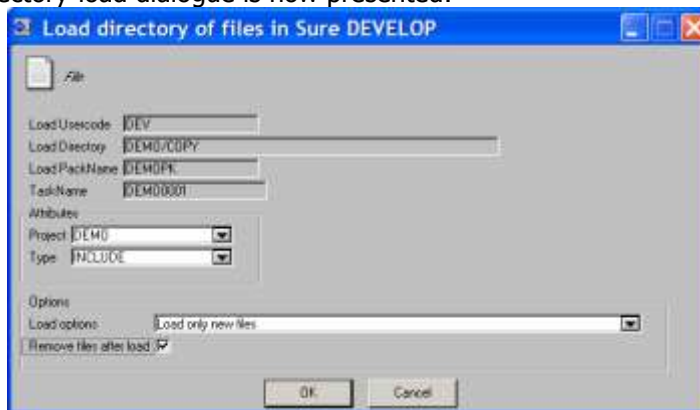
- ★ Right-click with the mouse on directory DEMO/COPY and choose 'Load files in Sure':



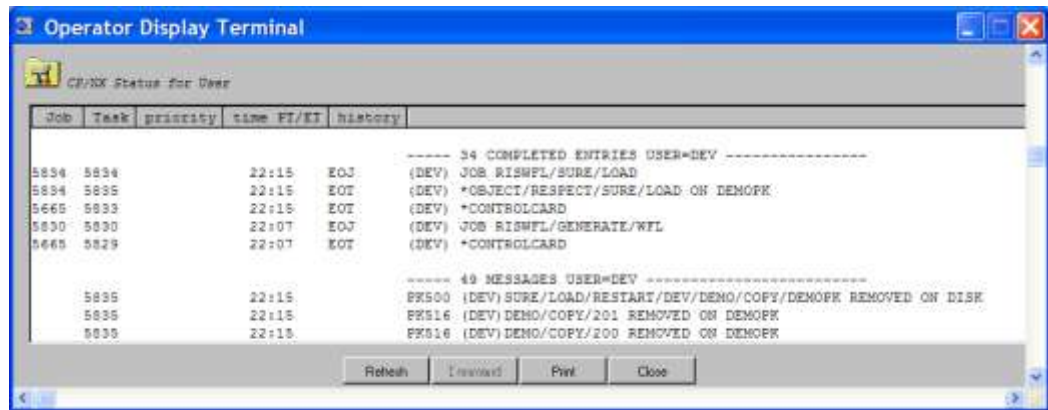
It is not possible to load files in SURE without connecting them to a task. Your current task (in this case DEMO0001) is pre-entered in the task-field, but via the drop-down box you can select another task. It is also possible to define a new task via button 'Define new task', or to connect yourself to another task from your Task-ToDo-list via button 'Search'.

- ★ Click on OK when you are satisfied about the task.

The directory-load dialogue is now presented:

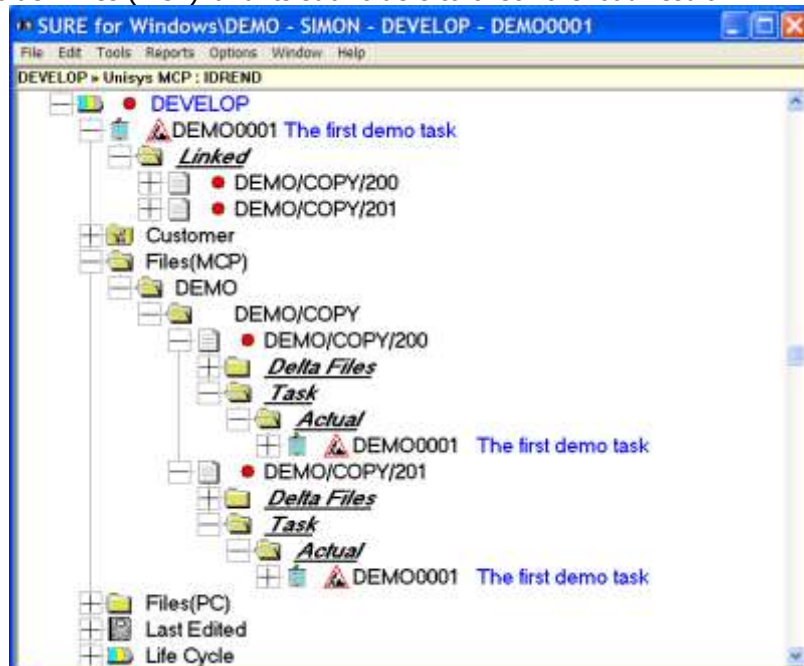


- ★ Enter the following load-attributes:
 - Project = DEMO (via the drop-down-box)
 - Type = INCLUDE (because we are going to load copy-files)
 - Load options = Load only new files (via the drop-down-box)
 - Set option Remove files after load
- ★ Click on OK to start the batch load process.
- ★ Double click on 'Status mix: DEV' to control the status of the load process:



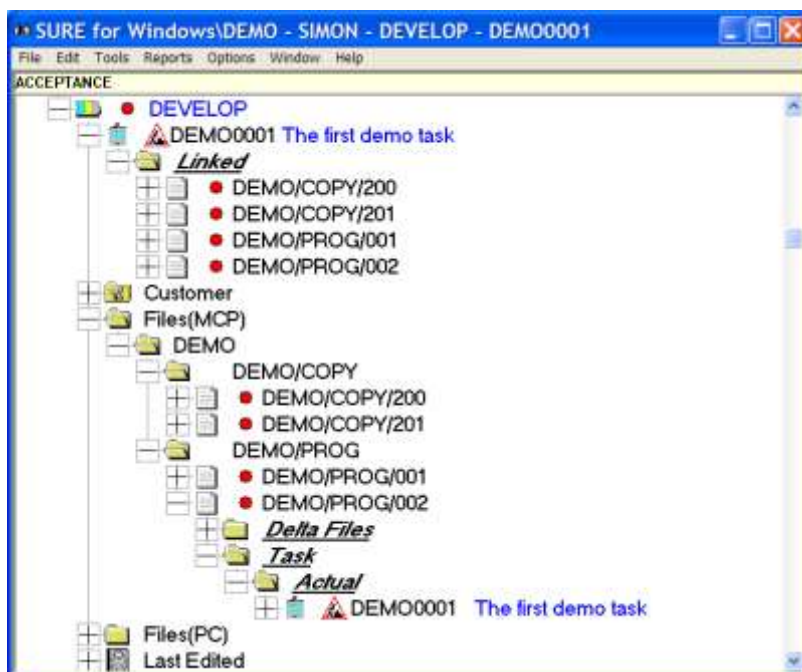
In this example screen the batch load process is just completed.

- * Open folder 'Files (MCP)' and its sub-folders to check the load result:



In this example you see that two files were loaded: DEMO/COPY/200 and DEMO/COPY/201. Both files are connected to the current task. The file-task connection is demonstrated in two ways: <files> are **linked** to <task>, and <task> is an **actual-task** of <file>.

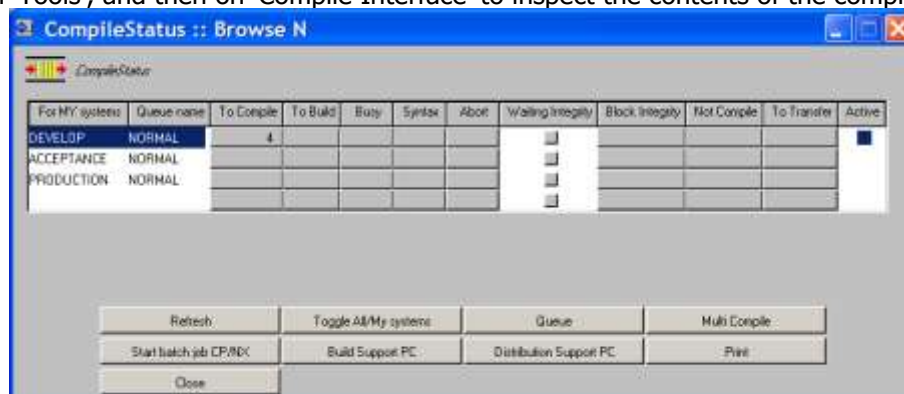
- * Repeat this actions for the second directory: DEMO/PROG/=
 - Open folder 'Unisys MCP <hostname>'
 - Open folder 'Sources (DEV)' on <work pack>:
 - Right-click with the mouse on directory DEMO/PROG and select 'Load files in Sure'.
 - Click on OK when you are satisfied about the task.
 - Enter the following load-attributes on the load-directory dialogue:
 - Project = DEMO (via the drop-down-box)
 - Type = SOURCE (because we are going to load program-files)
 - Load options = Load only new files (via the drop-down-box)
 - Set option Remove files after load
 - Click on OK to start the batch load process.
 - Open folder 'Files (MCP)' and its sub-folders to check to load result:



- Notice that all loaded files are marked with a red icon. This status icon has the same color as environment DEVELOP and it identifies the status of each source.
- * If you did not set option 'remove file after load' on the load dialogue, then please remove the two Cande directories from disk because the files are now loaded in SURE:
 - REMOVE SOURCE (DEV)DEMO/COPY/=
 - REMOVE SOURCE (DEV)DEMO/PROG/=

Files that are loaded in SURE are automatically placed into the SURE-compile-queue.

- * Click on 'Tools', and then on 'Compile Interface' to inspect the contents of the compile-queue:



There are 4 files in the 'To Compile Queue' of environment DEVELOP.


6 Build Support & Distribution Support

6.1 Compilation and Object files

Files that are loaded in SURE are automatically placed into the SURE-compile-queue. In this section, the SURE batch job is started. This job compiles programs via SURE, and deploys the object to the correct object-location.

The batch job performs the following functions:

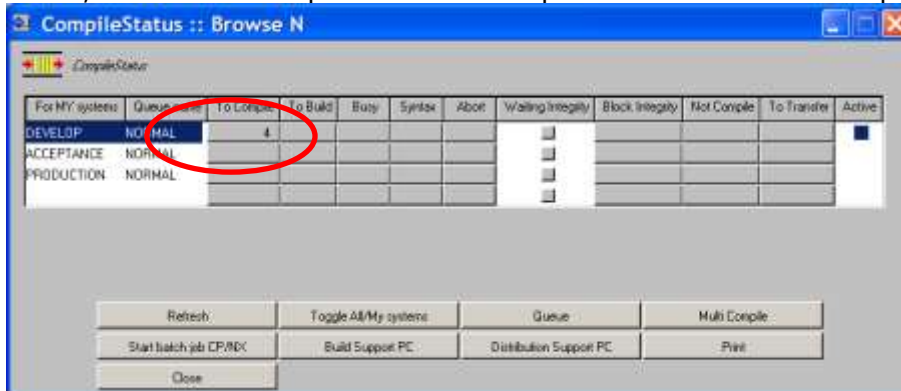
- A. Process the SURE compile queue: all files in the compile-queue are compiled. In case of an include file all referencing programs are compiled.
- B. Copy the compiled objects to the correct object location.
- C. Create delta-files between the changed files and their previous versions.
- D. Examine (scan) changed files for language statements that refer to other files.
- E. Process the SURE find queue.
- F. Process the SURE replace queue.

 **It may be possible that the programs that you are going to compile refer to one or more databases. In that case, the correct database description files must be made visible for usercode DEVBATCH. Otherwise the compilation process in step A fails.**

If you used the example programs DEMO/PROG/= and DEMO/COPY/= then you must copy descriptionfile (SURE)DESCRIPTION/INFDB AS *DESCRIPTION/INFDB (on your pack) and you must change the security of this descriptionfile to PUBLIC IO to make it visible.

 **The usercode where the SURE batch runs (DEVBATCH) must be a privilege usercode (PU), otherwise the copy process of step B fails.**

- * Click on 'Tools', and then on 'Compile Interface' to inspect the contents of the compile-queue:



In this example, the compile-queue for environment DEVELOP contains 4 programs.

- * Click on button 'To Compile' for environment DEVELOP:



The four files in the SURE compile queue.

- Click on one of the include-file names to name the file current (DEMO/COPY/200 in the example), followed by a click on button 'Open' to check the properties of that file.

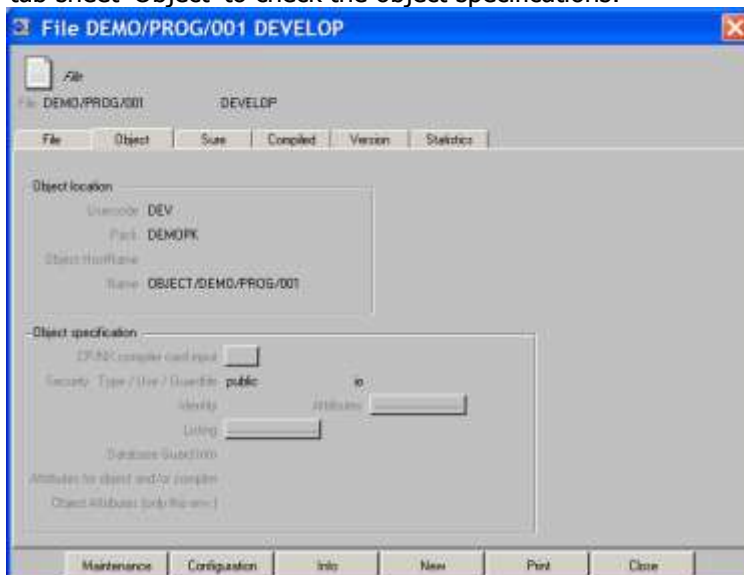


The file belongs to project DEMO and the file-type is INCLUDE.

- Click on tab sheet 'Object' to check to object specifications of this file.

This file does not have any object specifications, and that is correct, because the file is an include file and it can not be compiled.

- Click on button 'Close' to return to dialogue 'Compile details to-compile'.
Click on a program name (not an include file) to make it current (DEMO/PROG/001).
Click on button 'Open' to check the properties of that file.
Click on tab sheet 'Object' to check the object specifications:



The object of this file is (DEV)OBJECT/DEMO/PROG/001 ON DEMOPK. The object-location usercode and pack are inherited from the default object location of system DEMO as a result of option 'Files of this type inherit <object location 1> from their system' of file-type SOURCE.

- Click 2 times on button 'Close' to return to dialogue 'Compile Interface'.

- * Click on button 'Start batch job'



- * Set batch job options
 - 'Generate and Compile' (function A)
 - Transfer (function B)
 - Create delta files (function C)
 - Examine changed files (function D)

- * Click on button 'OK' to start the job:



A standard SURE evening job is available for each environment. The name of this job is WFL/<environment>/SURE, and the job was placed on disk during the installation.

- * Click on button 'View status batch' to control the status of the batch job:

Chapter 24 gives detailed information about the SURE compile + transfer process (functions A, B).
Chapter 18 gives detailed information about delta files (function C).
Chapter 26 gives detailed information about the examine process (function D).

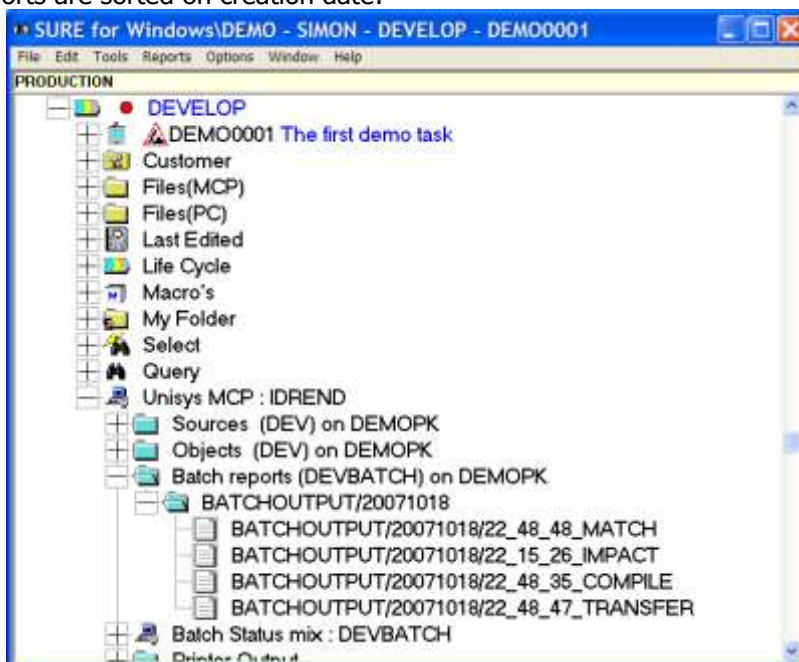
An important feature of SURE is the integrity mechanism: SURE controls that all objects in the run-time environment are compiled with the same set of copy-files. If a copy-file changes, then all programs that use that copy-file are recompiled.

Read chapter 24.12 through 22.12.5 for detailed information.

The SURE batch creates a number of overviews. These backupfiles can be accessed via the SURE browser via folder:

Unisys MCP <hostname> → Batch Reports <batch usercode>

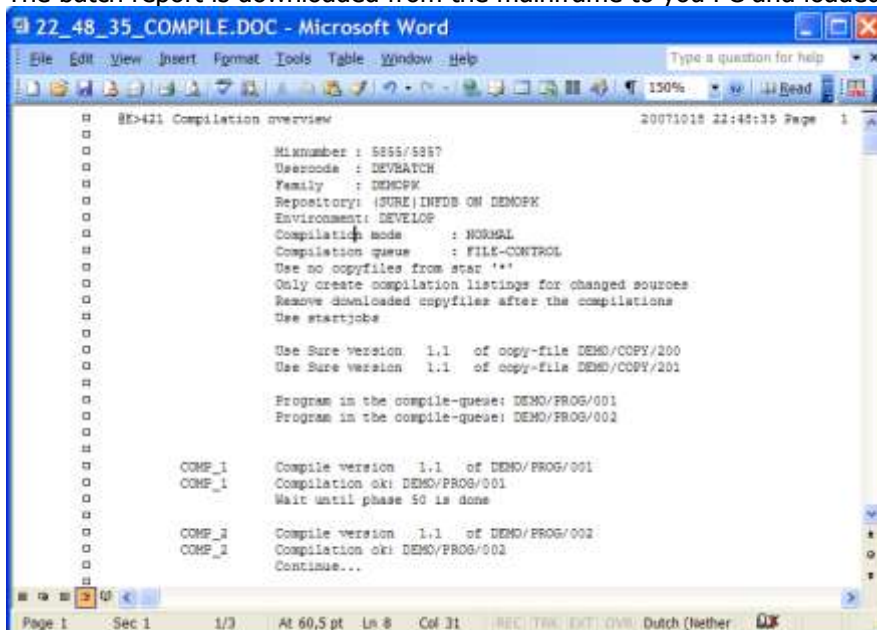
The reports are sorted on creation date:



The amount of days that batch reports are kept on disk can be defined via the environment properties → Tab: Sure Batch → field: Remove backupfiles after # days.

Open a batch report as follows:

- Right click on the compile overview BATCHREPORT\.....COMPILE
- Choose <Download>
- The batch report is downloaded from the mainframe to you PC and loaded in MS Word.



SURE supports a macro for MS-Word to view batch reports with 132 chars on a line.

6.2 Load object run statistics in SURE

It is possible to load run information from the system/sumlog into SURE. This run information is then for each file available in SURE.

Batch program RESPECT/SURE/LOG reads a system sumlog file, and stores the relevant information about programs runs from the sumlog file in SURE.

The procedure is explained via an example with file DEMO/PROG/002.

Create a small sumlog file that contains log info about several runs of object (SURE_DEV)OBJECT/DEMO/PROG/002:

- * Close the current sumlog file: ODT command TL
- * RUN (SURE_DEV)DEMO/PROG/002 ON <pack> (run it multiple times).
 - You can run the program via Cande but you can also run the program via the SURE explorer as follows: Right click on the source name → Miscellaneous → Run
- * Close the sumlog file again: ODT command TL

Check which sumlog files are resident: FILES *SUMLOG ON <sumlog pack>

The run information of the runs of DEMO/PROG/002 is stored in the last sumlog file in the list.

- * Load the sumlog file in SURE as follows:
RUN RESPECT/SURE/LOG;FILE LOG = <last sumlog file>

The sumlog information is presented via SURE as follows:

- * Right-click on file DEMO/PROG/002 and choose 'Statistics' from the menu:

| Timestamp | Usercode | Menu | Log | Hour | Object Created | File Version | Memory | Eq | Init PBit |
|---------------------------------|----------|------------|-----------|------|------------------------------------|--------------|--------|----|-----------|
| vrijdag 19 oktober 2007 7:55:47 | DEV | 6045/06050 | 1251 1111 | | donderdag 18 oktober 2007 22:48:46 | 1.1 | 2.635 | 0k | 15 |
| vrijdag 19 oktober 2007 7:55:47 | DEV | 6045/06047 | 1251 1111 | | donderdag 18 oktober 2007 22:48:46 | 1.1 | 2.745 | 0k | 18 |

The statistics screen shows one line with information per run of the object. Notice that the sumlog number is also stored in this run information (4th column). This makes it easy to find the correct sumlog if you want to know more details about a run of a program.

Program RESPECT/SURE/LOG also produces the following reports:

- Overview of wrong versions of objects (not compiled via SURE).
- Overview of illegal objects (not known in SURE).

Read chapter 29 of the SURE manual for more information about loading sumlogs into SURE.

6.3 Other batch procedures and backup procedures

The following batch procedures are available:

- Compiling and transferring objects (see before).
- Loading object run information into SURE (see before).
- Clean old files from pack (and backup them to tape) (chapter 28 of the manual)
- Backup all sources that are in maintenance by programmers (chapter 27 of the manual)
- Backup production objects (chapter 24.15.2)

The last three procedures are optional.

It is obvious that a correct backup procedure has to be installed for the database itself:
At least once a week a total database dump (off-line or on-line), and enough audit file backups to make it possible to rebuild the database from the last or the second last database backup tapes.

7 Program life cycle support

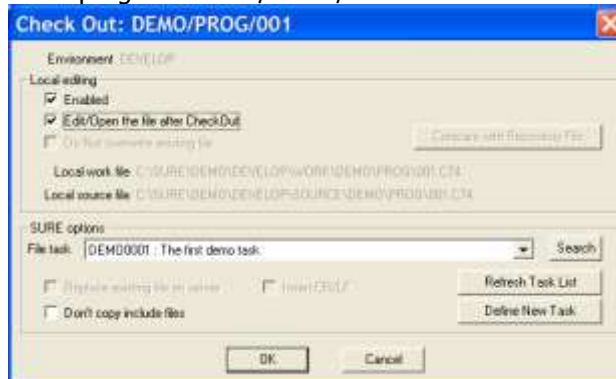
The following actions are done in this section:

1. Modify an existing file.
2. Transfer all modifications to the PRODUCTION environment
3. Add a new file and modify an existing file.

7.1 Modify an existing file

The procedure is explained via example program DEMO/PROG/001.

- * Right-click on program DEMO/PROG/001 and choose 'check-out' from the menu:



- Notice that 'local editing enabled' is set. This option enforces that the file is going to be modified on the PC.
- If local editing is enabled, the file is downloaded from the mainframe to a source directory on the PC. This is only done if the correct version of the file was not yet available in the local source directory. The file is then copied from the local source directory to the local work directory. The local source directory and the local work directory are defined via Menu-bar → Options → SURE options → tab: local options.

The names of the local source file and the local work file are mentioned on the checkout screen: it is the name of the local directory, followed by the name of the source.

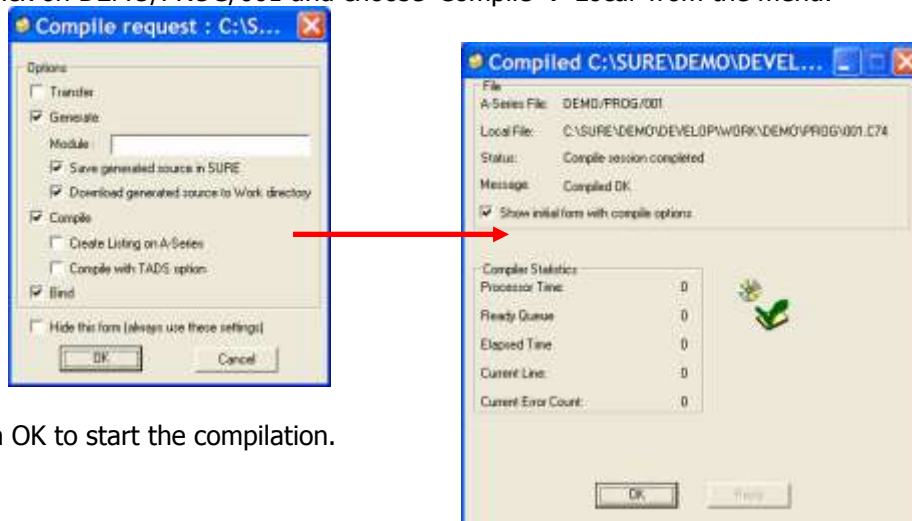
Notice the name of the task DEMO0001. The file is connected to this task, which means that the file is modified because of task DEMO0001. If necessary, it is possible to choose another task from the value-list.

- * Click on OK to confirm the 'check-out' options.
- * The file is now copied to the local work directory. Click on OK when the file is resident.
- * Click on OK to start the local editor:



By default, NOTEPAD is started as local editor. We made seamless interfaces to various other editors such as 'ED for Windows', 'MultiEdit', Programmers Workbench, etc. Please refer to chapter 12 of the SURE manual for detailed information about PC editing and local editors. You can choose another local editor via Menu-bar → Options → SURE options → tab: local options.

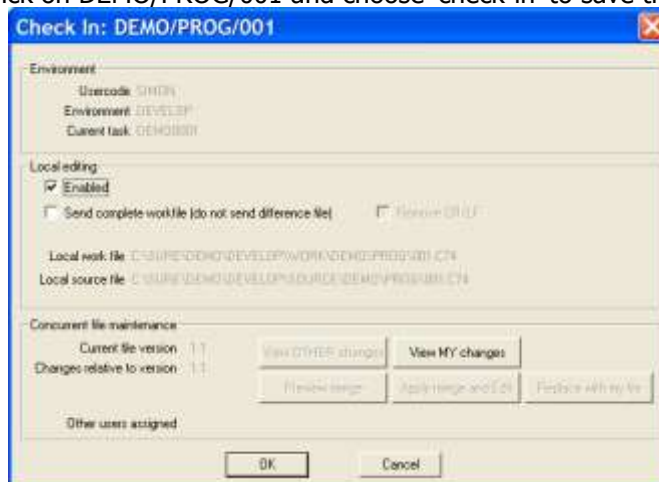
- * Make some small modifications in the file (add or modify a display, et cetera).
- * Save the file and leave the editor.
- * Right-click on DEMO/PROG/001 and choose 'Compile → Local' from the menu:



- * Click on OK to start the compilation.


The local work file is compared with the local source file and the differences are uploaded to the mainframe and merged with the original file on the mainframe. The merge-result-file is compiled on the mainframe. Syntax errors are downloaded to the PC, and can be merged into your local work file.

- * Right-click on DEMO/PROG/001 and choose 'check-in' to save the modified file in SURE.



- Notice button <View my changes>. This function can be used to compare the previous version of the source and the new version.

Function check-in works as follows: The local work file is compared to the local source file and the differences are uploaded to the mainframe and merged with the mainframe original-file. The result file is saved in the repository. The local work file is moved to the local source directory, and becomes local source file.

- * Click on the status icon of DEMO/PROG/001 to show the file-log:  DEMO/PROG/001



- A history of actions on this file is logged in SURE.
- Another method to view the log is: Right click on source name → Log

★ Right-click in DEMO/PROG/001 and choose 'Miscellaneous → Run' to run the program:



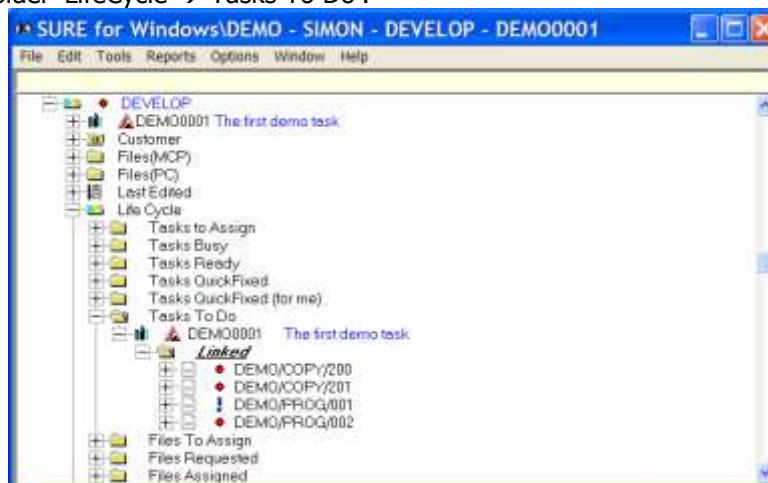
In this example screen the objects just completed.

7.2 Transfer modifications to the next environment

All adaptations are linked to a task: the current task of the programmer who made the adaptation. When the programmer has finished his activities for the task, then that task can be transferred to the next environment. All sources that were adapted because of the task are then transferred too.

Up until now, we always worked for the same task: DEMO0001. All loaded files are linked to this task, and the program that was modified is obviously also linked to this task.

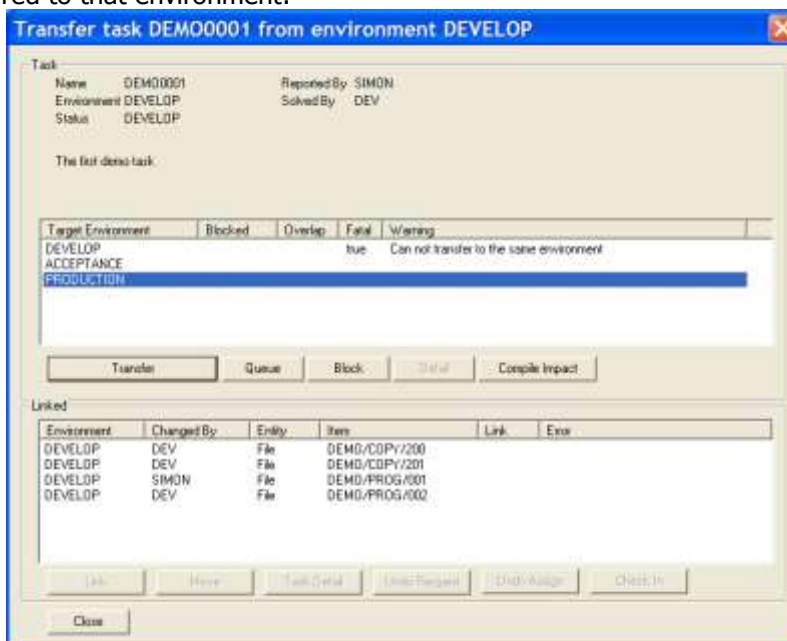
★ Open folder 'LifeCycle → Tasks To Do':



★ Right-click on your current task DEMO0001 and choose 'Transfer' from the menu.

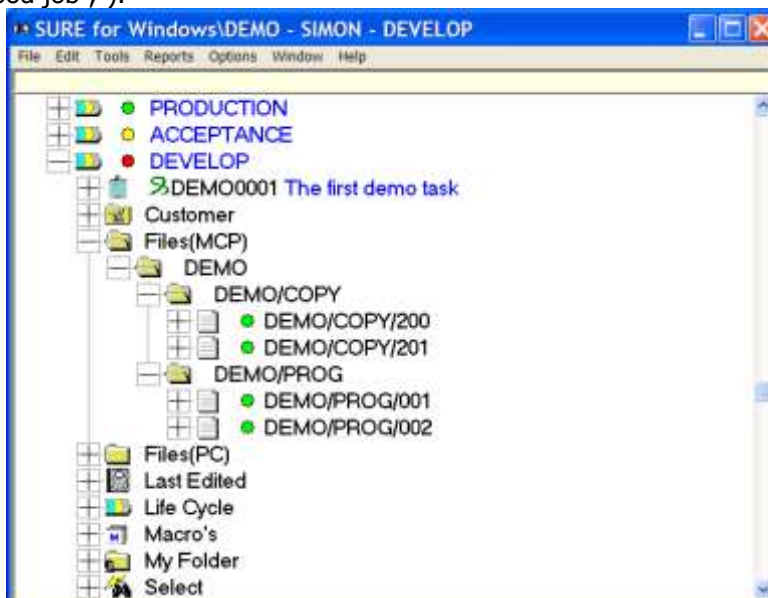
A transfer-dialogue is presented where the possible destination environments are given, and where all files are listed that are modified because of the task.

- * Click on the line with destination environment PRODUCTION, because the task has to be transferred to that environment:



- * Click on the button 'Transfer' to transfer the task from DEVELOP via ACCEPTANCE to PRODUCTION.

The status of the task is now changed to SOLVED. The icon just before the taskname represents this status: DEMO0001. The small green curling sign is used in Holland at kindergarten when you did a good job ;-).



- Notice the color of the status-icon of the DEMO/= sources. The color is changed to green (the color of environment PRODUCTION), which means that the version of the source in environment DEVELOP is equal to the version in environment PRODUCTION.
- * Right-click on task DEMO0001 and choose 'Properties'. The status of the task is now SOLVED.

- ★ Right-click on file DEMO/PROG/001 and choose 'Status':



FileVersion 1.2 is now active in all environments.

Read chapters 38.8.1 to 38.8.12 for detailed information about transferring a task.

7.3 Add a new file and modify an existing file

The next step of the demonstration is to add a new file and to modify an existing file. These actions are only possible if you have a task to work on. In the previous paragraph the current work task was transferred to environment PRODUCTION, and got status SOLVED. Therefore, we first have to add a new task, and define it as our current work task.

- ★ Enter a new task and define that task as 'my current work task' (see the procedure in chapter 4).

The new task is called DEMO0002.

- ★ Modify existing file DEMO/PROG/001 following the procedure as described in paragraph 7.1

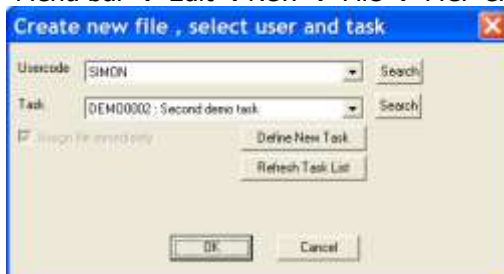
Notice that DEMO/PROG/001 is linked to task DEMO0002.

- ★ Right-click on DEMO/PROG/001 and choose 'Status' from the menu:



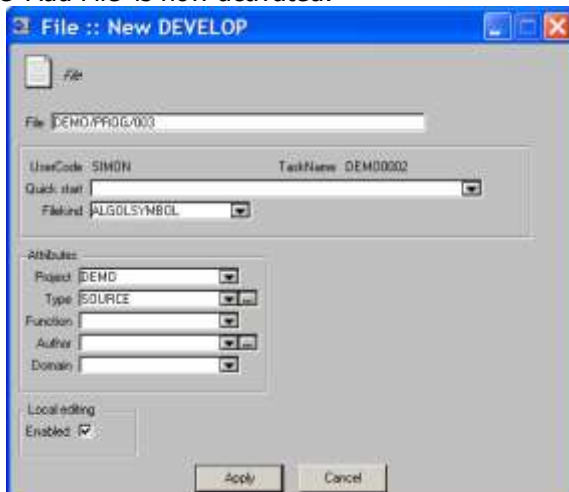
- The FileVersion of the program is now 2.1 in DEVELOP and 1.2 in ACCEPTANCE and PRODUCTION.

- ★ Click on 'Menu bar → Edit → New → File → MCP environment' to add a new file:



- ★ Click on 'OK' to confirm the current task.

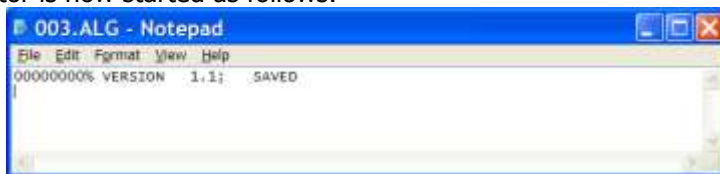
Dialogue 'Add File' is now activated:



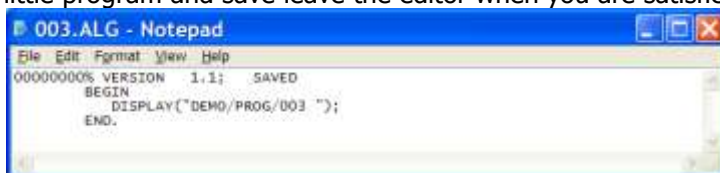
In this case we create an Algol program:

- * Enter the following file attributes:
 - File Name = DEMO/PROG/003
 - FileKind = ALGOLSYMBOL
 - Project = DEMO
 - Type = SOURCE
 - Set option 'Local editing enabled'
- * Click on button 'OK' to add the new file name in SURE.
- * Reply question 'Start Editor' with YES.

The editor is now started as follows:

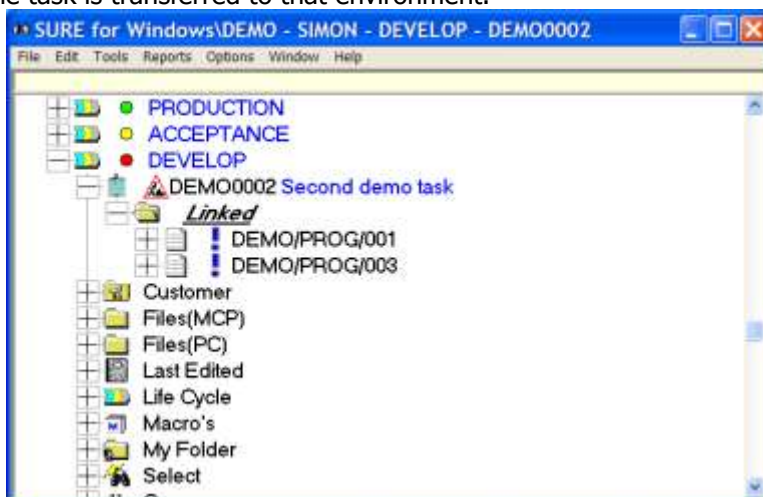


- * Write a little program and save leave the editor when you are satisfied.



- * Right-click on files DEMO/PROG/3 and choose 'Check-In' to save the file in SURE.

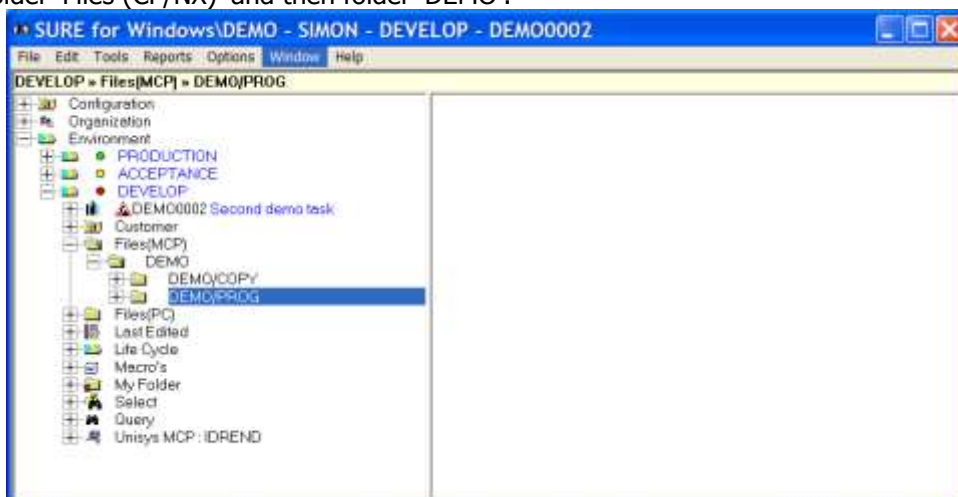
Both files are now connected to the same task. The modifications are transferred to production when the task is transferred to that environment.



7.4 Inquire and browse functions

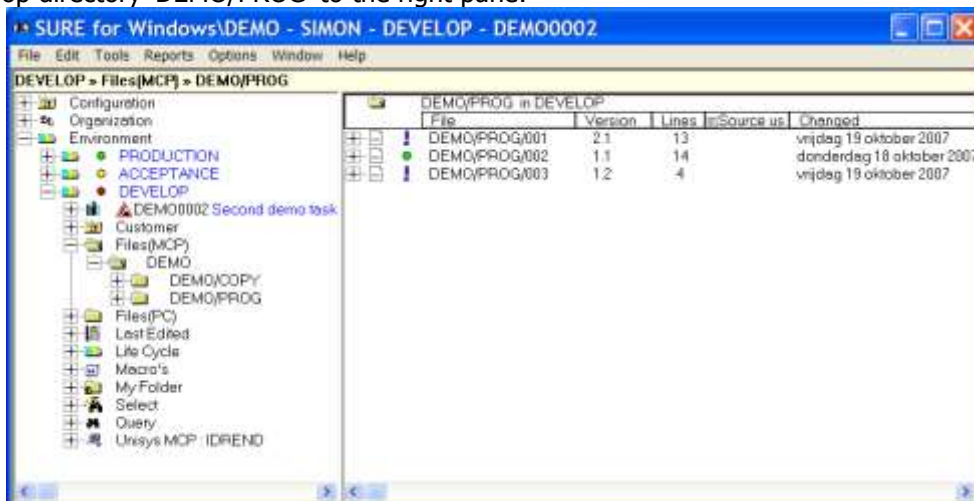
Log-on to SURE with usercode

- * Open folder 'Files (CP/NX)' and then folder 'DEMO':



- Click on Menu bar → Window → New, to open a right pane.

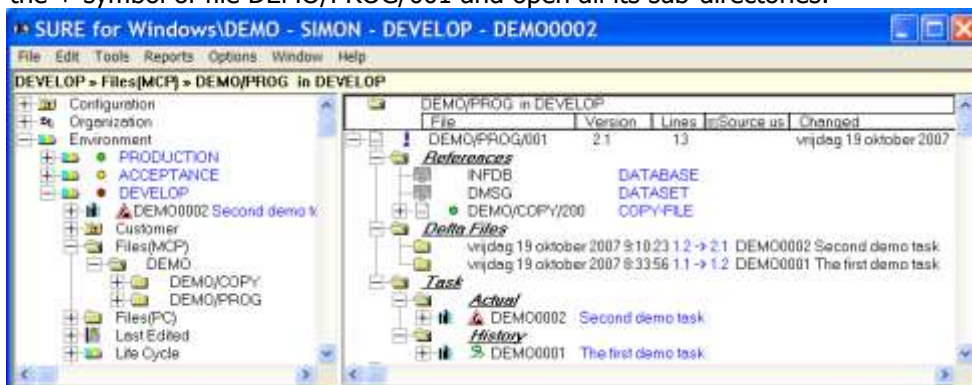
- * Drag/drop directory 'DEMO/PROG' to the right pane:



Notice the columns 'version', 'lines', 'source-usercode' and 'changed'. The information in these columns is retrieved from the SURE repository on the mainframe. It is possible to change the columns to your own needs via 'Menu bar → Options → Attributes → Files'. A screen appears where you can choose the information that you want to see in the columns.

It is possible to sort the browse-list per column.

- ★ Click on the + symbol of file DEMO/PROG/001 and open all its sub-directories:

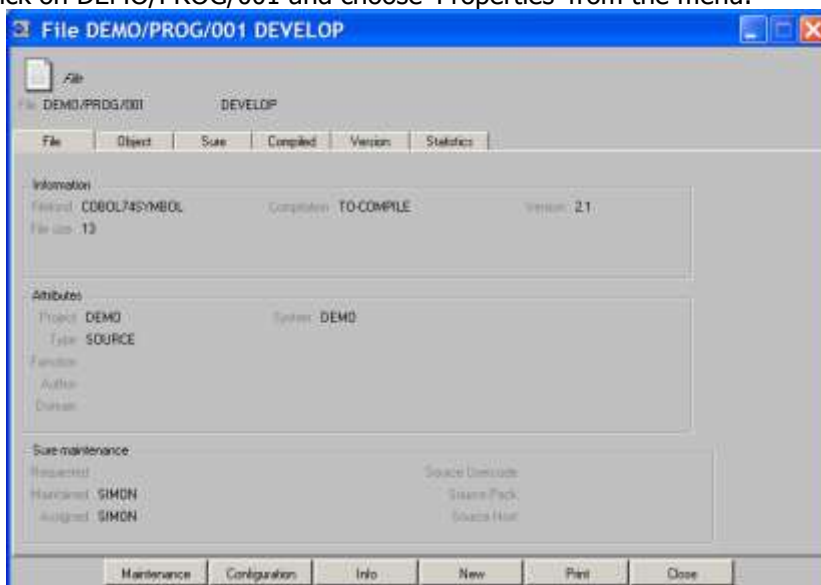


The file was modified in the past because of task DEMO0001 (sub-directory task/history). Tasks that were linked to the file for maintenance purposes are automatically loaded in this history when they get status solved.

The file references one copy-file and two datasets.

A click on the + symbol of copy-file DEMO/COPY/200 gives all programs referencing this copy-file.

- ★ Right-click on DEMO/PROG/001 and choose 'Properties' from the menu:



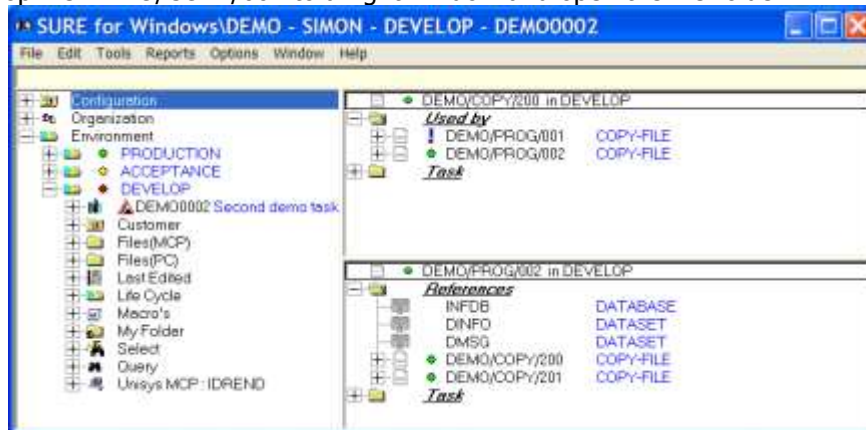
- ★ Click on button 'Info' and choose 'List' to see the contents of the file.
- ★ Click on button 'Close' to return to the SureBrowser.

8 Support functions

8.1 Cross reference and data dictionary functions

Each time a new version of a source is loaded in SURE, the source is cross examined for used datasets, sets, databases, libraries, input or output files, external procedures, et cetera. All found issues are linked to the source. The examine function is normally done during the evening batch, but on-line command EXAMINE is available for direct results.

- * Drag/drop file DEMO/COPY/001 to a right window and open the file-folder:



Open a right window via Window → New
Copy file DEMO/COPY/200 is in use by two programs.
Program DEMO/PROG/002 references two copy-files.

It is not only possible to inquire the attributes of a file; it is also possible to select groups of files using their attributes. A special QUERY facility is available where groups of files can be selected.

- * Double click on 'Query' to enter the query function.
- * Click on button 'Advanced'



Example 1: Select all files that are placed in the SURE evening compile queue:
Enter in 'Advanced expression' : COMPILE-STATUS(TO-COMPILE)

Example 2: Select all files that are maintained by usercode <your usercode>:
Enter in 'Select Criteria' : MAINTAINED(<your usercode>)

Example 3: Select all files that use copy-file DEMO/COPY/200:
Enter in 'Select Criteria' : COPY-FILE(DEMO/COPY/200).

Read chapters 14 and 26 for detailed information about the select function.

8.2 Delta files

All adaptations that are made to sources are kept in the repository. After an adaptation to a source SURE makes a comparison between the previous FileVersion that was active in an environment, and the newly saved FileVersion. The result of such a comparison is called a DELTA file, and these delta files are stored in the repository.

It is possible to browse through delta files, or to roll a file back to an old version using these delta files.

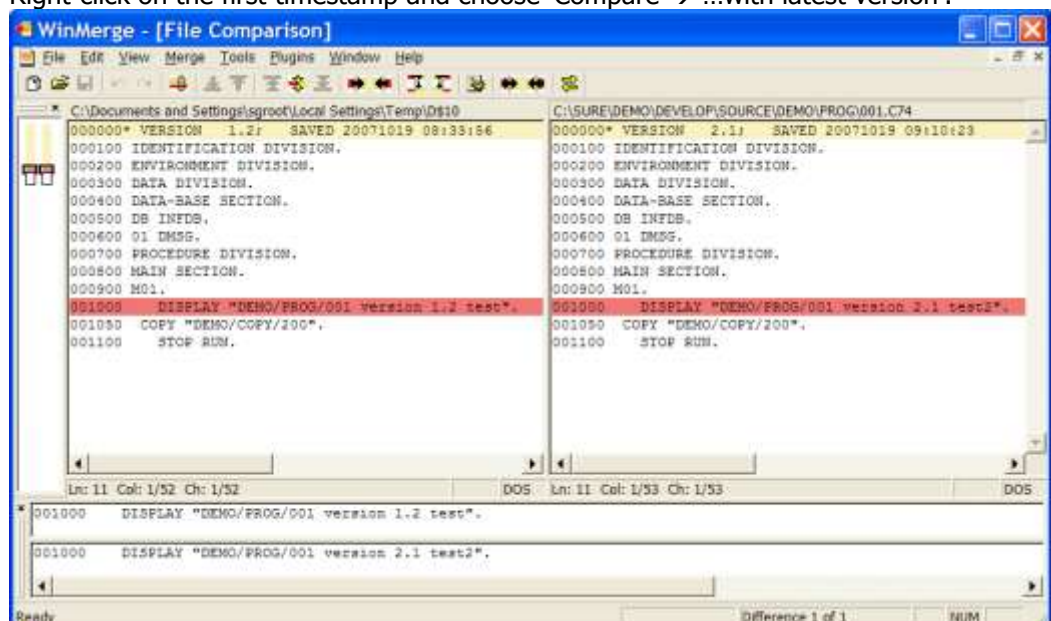
- * Open folder DEMO/PROG/001 and then 'Delta-files':



Double click on a line with a delta-file timestamp to show the contents of that delta-file.

It is also possible to compare two versions of the source:

- Right-click on the first timestamp and choose 'Compare -> ...with latest version':



- The two versions are loaded in a shareware tool WinMerge, and the differences are marked.

It is possible to compare all old FileVersions.

Refer to chapter 18 of the SURE manual for detailed information about delta files.

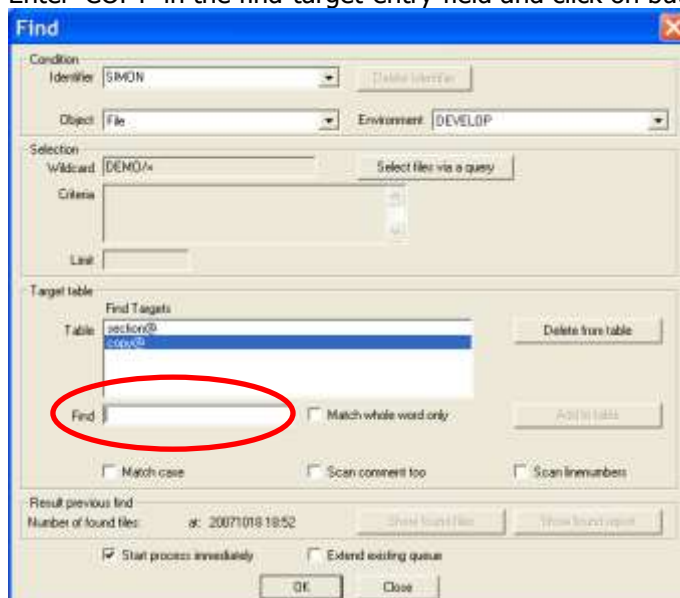
8.3 Find and replace utilities

The FIND and REPLACE utilities in SURE can be used to find/replace one or more targets in a group of sources.

The FIND utility is explained via an example. The procedure of the REPLACE utility is similar to the procedure of the FIND utility.

Suppose you want to find the words 'SECTION' and 'COPY' in COBOL programs.

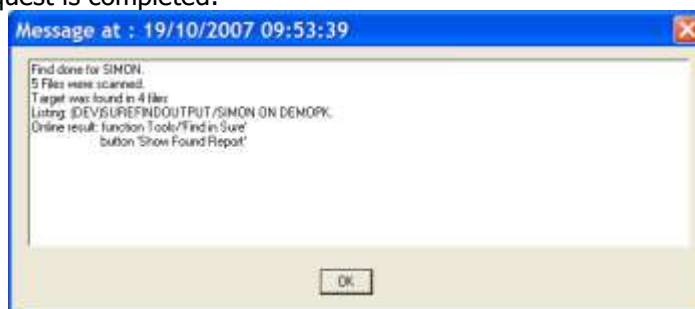
- * Click on 'menu-bar → Tools → Find' to enter the find-utility.
 - Click on button 'Query'
 - o enter 'DEMO/=' in the Wildcard field
 - o Click on button <Advanced> to enable the lower part of the query-screen.
 - o Click on <OK>: the query parameters are copied to the find-screen.
 - Enter 'SECTION' in the find-target-entry-field (red circle) and click on button 'Add to table'
 - Enter 'COPY' in the find-target-entry-field and click on button 'Add to table':



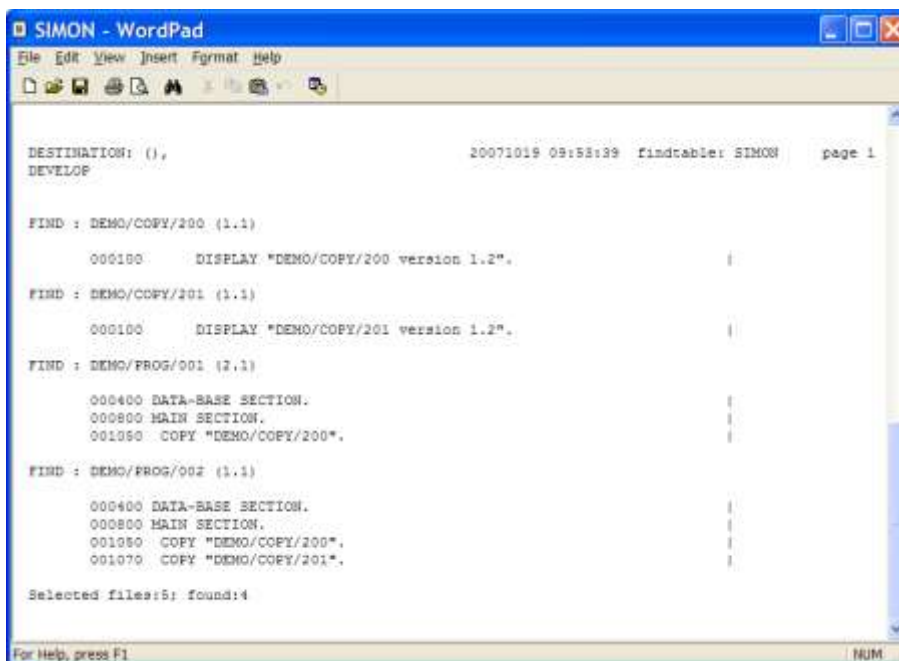
Wildcard 'DEMO/=' means that all files starting with that prefix are placed in the find-queue. The find-table consists of the find-targets 'COPY' and 'SECTION'. All files in the find-queue are scanned for those targets.

- * Click on OK to start the find utility on the mainframe.

The sources are scanned on the mainframe. After a short while you get a notification that the find-request is completed:



- * Open the FIND dialogue again and click now on button 'Show Found Report' to see the find-results:



```
SIMON - WordPad
File Edit View Insert Format Help
DESTINATION: ( ), 20071019 09:58:39 findtable: SIMON page 1
DEVELOP

FIND : DEMO/COPY/200 (1.1)
000100 DISPLAY "DEMO/COPY/200 version 1.2".

FIND : DEMO/COPY/201 (1.1)
000100 DISPLAY "DEMO/COPY/201 version 1.2".

FIND : DEMO/PROG/001 (2.1)
000400 DATA-BASE SECTION.
000800 MAIN SECTION.
001000 COPY "DEMO/COPY/200".

FIND : DEMO/PROG/002 (1.1)
000400 DATA-BASE SECTION.
000800 MAIN SECTION.
001000 COPY "DEMO/COPY/200".
001070 COPY "DEMO/COPY/201".

Selected files:5; found:4

For Help, press F1 NUM
```

Chapter 20 of the SURE manual gives detailed information about find and replace.

9 Epilogue

This concludes the SURE Quick Start Demonstration. We thank you very much for your time and attention and, above all, your interest in the SURE 'change & control' system.

We hope that these examples have given you a basic understanding of SURE. Many other features of SURE (which are not mentioned in this short demonstration), are available to support the developers and change-control staff in their daily jobs.

For detail information concerning SURE, consult the SURE reference manual.

If you have any additional questions about SURE, please do not hesitate to contact us.

Once again, thank you and good luck with SURE.

To execute the Quick Start Demonstration again, simply reload the SURE files from the demonstration tape to your system as described in chapter 2.