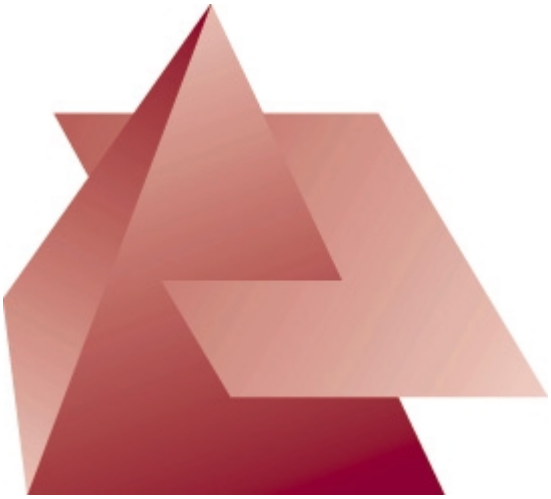
CASEDOK™ RUNNER for COOL:Gen

Version 4.0



This manual has been created with the **CASEDOK™ DEVELOPER** workstation.

Table of Contents

1 Contact Deron

Address: **Deron AG**
Webergutstrasse 4
3052 Zollikofen
Switzerland

Phone: **+41 31 910 44 55**
Fax: **+41 31 910 44 49**

Email: **info@deron.com**

Internet: **www.deron.com**

In USA and CANADA, contact our partner:

Address: **CANAM Software Labs, Inc.**
200 Matheson Blvd. West Suite 103
Mississauga, Ontario
Canada L5R 3L7

Phone: **(905) 712-8985**
Fax: **(905) 712-0043**

Email: **support@canamsoftware.com**

Internet: **www.canamsoftware.com**

2 What's new in version 4.0

The following changes occurred since version 3.0:

- Improved user interface, adapted to Windows 2000 look and feel
- Support for the scheduling of Runner job execution at a given time or period
- Support for COOL:Gen 6.0 schema release 9.0.A2
- New ASYNC statements in action diagrams
- Window diagrams adjusted to different display resolutions
- Long, expressive report definition file names
- Support for the organization of report definitions in any folders
- More flexible definition of subroutines
- Improved support for headings of any level
- Bug fixes in table output

Release History

Version 3.0

- Easier to use Runner interface
- Ability to directly select objects included in the report (not with CE)
- Improved, pixel-exact GUI reports
- New selection and sorting capabilities in the reports through CASEDOK Developer
- Availability of report parameters in conjunction with CASEDOK Developer
- Immediate access to COOL:Gen 5 workstation models after installation
- Output adjusted to paper size of current default printer
- New plain text output driver

Version 2.2

- Support for COOL:Gen version 5 and the metamodel 8.0.A3
- Year 2000 compliance
- Various bug fixes

Version 2.1

- COOL:Gen 4 and schema 7.0.A8 support
- Action Diagramming
- Workstation model support

3 What is CASEDOK RUNNER?

CASEDOK RUNNER is a fully-automated repository publisher for COOL:Gen designed to produce high-quality reports. The tool extracts model data directly from the Central Encyclopedia (CE), the Client/Server Encyclopedia (CSE) or the local workstation files (DAT files) and produces text and hypertext documents in various formats such as Rich Text Format (RTF), Windows Help, HTML or Plain Text. A standard report library which is delivered with the CASEDOK RUNNER tool covers all phases of the software life cycle from planning to implementation. Different kinds of reports support people in their daily work with the COOL:Gen platform. The report library includes reports that may be helpful for developers, analysts, QA specialists, project managers or even end users.

CASEDOK RUNNER allows you to define a documentation job including one or more reports. The objects included in a report can be selected using names or wildcards (*,?). Job settings, such as selected report(s), model name and publishing format(s), can be saved and reloaded for later runs. The jobs are seamlessly executed and as an option also in batch (only for the CE). For the Central Encyclopedia (CE), the client-host interaction is controlled by a customizable terminal script which you can adapt to your site-specific environment.

CASEDOK Runner also includes a macrofile concept which allows you to substitute original COOL:Gen terms by individual names. In addition to COOL:Gen object and property names, titles, fixed strings and property values can be translated (e.g. Domain='B' into 'Basic' or 'Elementary Process Indicator=Yes' into 'Elementary Process'). Any term or string appearing in a document can be adapted to individual needs or translated to foreign languages.

CASEDOK RUNNER is very easy to use! This tool lets you produce well-structured, high-quality documents 'at your fingertips'. CASEDOK RUNNER simplifies the communication with end users and helps you save time and money in your COOL:Gen projects.

CASEDOK RUNNER is one of the products based on the CASEDOK platform. Other products are **CASEDOK DEVELOPER**, **CASEDOK WALKER** and **CASEDOK WebEXPLORER**. With CASEDOK DEVELOPER you can create your own individual reports. This powerful workbench lets you access the full range of metamodel objects and provides a rich set of markup elements such as paragraphs, tables, lists, index entries, bitmaps, hyperlinks, and a lot more. You can even embed native RTF or HTML code. CASEDOK WALKER is an encyclopedia walker to inspect model information, for example to find corrupt data. This tool can also help you in developing reports. With CASEDOK WebEXPLORER you can make your model available on the intranet. People can use their favorite web browser to navigate online through their models directly on the database.

4 Installation and Configuration

4.1 Technical Requirements

Windows 95/98/NT/2000	
Hardware	Pentium II/300MHz or higher, 64MB Memory, 30MB free disk space
Operating System	Windows 95/98 or NT 4.0 or Windows 2000
Tools	Windows Help Workshop 4.0
Webbrowser	Netscape 4.0 or MS Internet Explorer 4.0 or higher
Encyclopedia	CE, CSE or local workstation files (Read-Only API)
Metamodel release	9.0.A2; prior releases with some limitations)
CSE Database	DB2, Oracle, Informix Online, MS/SQL Server, DB2/2, Nonstop SQL*
CSE Database Access	ODBC (32bit)
Seamless Communication to Host	Personal Communications, GWTEL, or any other 3270 emulator with 32bit HLLAPI interface.
CE/MVS Host Environment	TSO, QMF, REXX

* database version is dependant on the local platform on which the server will reside

4.2 Authorization Requirements

- **Database access:**
  The CASEDOK user must have **READ access for the COOL:Gen schema and data tables**. This authorization has to be defined on database level. Ask your database administrator for help.
- **Model access:**
  A CASEDOK user must also be defined as COOL:Gen user. Ask your encyclopedia manager for help, if you have not yet the appropriate rights.
- **Access to local workstation files:**
  No authorization check.

4.3 Installation steps

The following table shows the steps to a successful installation of the CASEDOK RUNNER product for different platforms.

CASEDOK Runner for CSE and local DAT files:

Steps	**Refer to**
Installing the client software	chapter 4.4: Client installation
Setting the configuration parameters	chapter 4.5: CSE configuration parameters

CASEDOK Runner for CE/MVS:

Steps	**Refer to**
Installing the client software	chapter 4.4: Client installation
Installing the server software	chapter 4.6: CE/MVS server installation

4.4 Client installation

The CASEDOK installation is done by an InstallShield script, which leads you through the installation procedure. Depending on the product license code, one or more CASEDOK components are installed. After installation, you will find the CASEDOK program(s) in the Windows Start menu.

Start the CASEDOK software installation from diskette or CD with:

A:\SETUP.EXE or [d:]\DISK1\ SETUP.EXE

Product Code:	Product code that is delivered with the CASEDOK software: <xxxx-xxxx>
Product Selection:	CASEDOK software modules to be installed. For a local installation select both the kernel module <CASEDOK> and the <RUNNER> module.
Destination Directory:	Name of the directory where the CASEDOK software is to be installed.

Local DAT files:

To access to the local workstation files, you need the **COOL:Gen 6.x** (or COOL:Gen 5.x with some limitations) toolset. Please verify that the COOL:Gen 6.x (or COOL:Gen 5.x) directory appears in the path environment variable.

Windows Help Compiler:

If you want to produce Windows Helpfiles, you must install the **Microsoft Help Workshop 4.0.** It is a part of the Windows SDK delivered with Microsoft Developer Studio or other development tools. If you are not sure about the existance of the help compiler on your system, look for the HCW.EXE file.

4.5 Configuration of CSE and local DAT files

To access a COOL:Gen Client/Server repository, CASEDOK uses dynamic SQL statements. The software configuration is done on the CASEDOK client; there is no server part to be configured.

CASEDOK stores all the required configuration parameters in the CASEDOK.INI file, which is located in the CASEDOK working directory. To change the configuration parameters, select **<View> <Options>** from the menu bar of the CASEDOK Runner window.

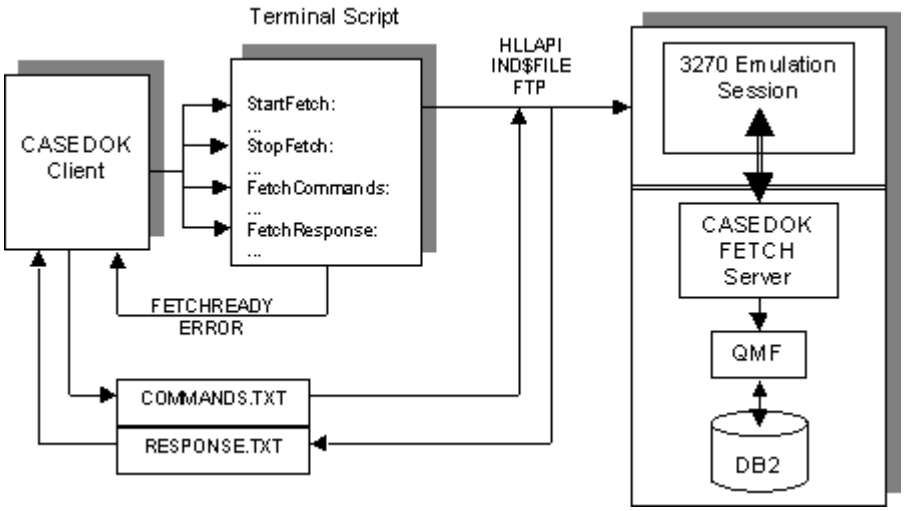
Parameter	Register Tab	Description
Macros	Reports	Location of the CASEDOK macrofile: <d:><path><filename>. The file contains the macros used in the CASEDOK reports. Macros are customizable text variables such as titles, object names or property values. For more information please read the description in the macrofile.
DocSuper	Reports	Document output directory where generated documents are stored. This may be a local drive or a netdrive.
RepositoryInterface	Repository	Select <ODBC> for ODBC access to Oracle, Informix, MS/SQL or other CSE databases. Select <Workstation API> to access the local DAT files.
ModelPath	Repository	Model path for the local DAT files (Workstation API). If this parameter is empty, the default COOL:Gen model path is used.
SQLDBName	Repository	Database alias for the ODBC connection to the CSE database. This parameter is not used for access to local DAT files.
SQLTABP	Repository	Table prefix for the CSE database tables (database owner). This parameter is not used for access to local DAT files.
SQLENCUSER	Repository	User ID for COOL:Gen (controls the model access). Attention: This value may be case-sensitive! Note: This user ID may be different from the user ID you enter when logging on to the repository database. This parameter is not used for access to local DAT files.

SQLBlockSize	Repository	Limit for the number of objects expanded in a single query. A larger block size increases the performance, but needs more resources at a time. Reduce this limit, if you get a maximum object size error during a documentation run. This parameter is not used for access to local DAT files.
SQLCacheSize	Repository	Number of objects to be stored in the local cache. This parameter is not used for access to local DAT files.
BICOKOUT	Debugging	If a filename (<d:><path><filename>) is specified, extensive message logging is written to this file. Activate this logging option only for debugging purposes, because the performance will strongly be affected.
SQLTrace	Debugging	Switch ON to enable a SQL trace. The trace data will be written to the BICOKOUT logfile. Activate this logging option only for debugging purposes, because the performance will strongly be affected.

4.6 CE/MVS server installation

4.6.1 Overview: Client - Host interaction

Due to the flexibility which CASEDOK offers for accessing the CE/MVS repository, it's customization needs some understanding of how CASEDOK operates. The following diagram shows how the components work together.



Client and Server

To access CE/MVS repository data CASEDOK uses a REXX procedure called FETCH, which runs as a server on the host. The FETCH procedure uses QMF to pull data out of the DB2 repository database. **Note:** FETCH may be customized to use an alternative query tool. CASEDOK's ' XDBI interface' is the way to do that.

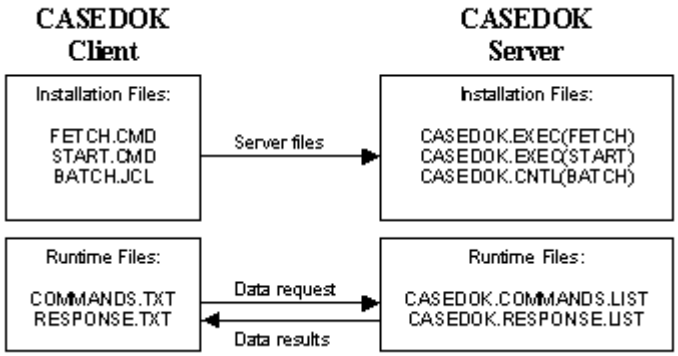
The communication between the CASEDOK client and the server procedure FETCH happens via a 3270 emulator session, which is accessed by the HLLAPI interface provided with most communication software. The HLLAPI interface is used for the flow control with the terminal and for the file transfer. The entire communication between the CASEDOK client and the FETCH server is encapsulated in a customizable script. **Note:** CASEDOK works with the EHLLAPI or the WinHLLAPI version. For communication software where send/receive file commands are not available, DDE can be used together with the software's macro language and e.g. FTP put/get file commands.

Work modes

The FETCH server is able to continuously deliver extracted data to the CASEDOK client, or to collect data in a single large response file. In first case, FETCH is an online TSO task, whereas in second case FETCH can be a batch job. A REXX script to start FETCH in foreground and a sample JCL to run FETCH in batch mode is provided.

Transfer files

The setup routine installs the CASEDOK server files on the client. The server files must be sent to the host and (before or afterwards) customized to site-specific parameters. Once installed, the CASEDOK application uses two files for data requests and data results (Commands and Response).



Accounts

A CASEDOK user requires a TSO terminal session. In this TSO account the FETCH server allocates the working and data files. If there are several CASEDOK users, it may be convenient to install the FETCH server once in a central account. You may also decide to install it in each individual TSO account.

4.6.2 Server installation

To install the CASEDOK server programs on the host, you have to do the following steps:

Step	To do	Where
1	Allocate the CASEDOK server datasets on the host (public or private)	TSO or ISPF
2*	Upload the CASEDOK server software to the host	Emulator or DOS window
3	Configure the site-specific parameters in the REXX script(s)	"START.CMD" and/or "BATCH.JCL" file
4	Customize the terminal script(s) for seamless mode (forground or batch)	Terminal script "FETCH.TSC" and/or "EXAMPLE_BATCH.TSC"
5	Configure the CASEDOK runtime parameters for CE/MVS	CASEDOK Runner window: <VIEW><SETTINGS> (CASEDOK.INI file parameters)

* You can either upload the standard server files to the host and configure them with a host editor; or you can edit the server files on your PC and upload the configured files to the host.

4.6.2.1 Allocating the CASEDOK server dataset(s)

CASEDOK uses it's own datasets to store the server programs. You have to decide now, wether you want to install the CASEDOK server once per user or just once per site. Once per user means that you allocate and install the CASEDOK server datasets in the individual user TSO account(s). If you want a central installation, the CASEDOK server datasets must be available in a public library.

Please allocate the following CASEDOK server datasets on your mainframe. The CASEDOK.CNTL dataset is only required if you want to use the batch mode.

```
ALLOC DA(CASEDOK.EXEC) RECFM(V B) LRECL(255) SPACE(80,200) BLOCK(3120) DIR(5)
```

```
ALLOC DA(CASEDOK.CNTL) RECFM(F B) LRECL(80) SPACE(10,10) BLOCK(3120) DIR(2)
```

4.6.2.2 Uploading the CASEDOK server

Upload the CASEDOK server programs "START.CMD" and "FETCH.CMD" to the allocated CASEDOK.EXEC dataset. Upload the "BATCH.JCL" file to the CASEDOK.CNTL dataset for batch. The appropriate DOS commands for a 3270 session filetransfer (IND$FILE) look like this:

```
SEND START.CMD A:CASEDOK.EXEC(START) ASCII CRLF
SEND FETCH.CMD A:CASEDOK.EXEC(FETCH) ASCII CRLF
SEND BATCH.JCL A:CASEDOK.CNTL(BATCH) ASCII CRLF
```

Use the appropriate commands for other types of filetransfer (e.g. FTP). The format must be ASCII with CRLF (carriage return/line feed).

4.6.2.3 Configuring the REXX script

You have to customize the server program "START" which is a REXX procedure to start the FETCH server in forground. The following 6 parameters (bold lines) must be adapted to site-specific terms:

```
/*************************************************************
** CONFIGURATION                                            **
** You have to customize the procedure for your site.       **
** Replace the following parameters with your site          **
** specific values:                                         */

qmfclst="QMF.330.DSQCLSTE"         /* QMF Clist Module      */
qmfexec="QMF.330.DSQEXECE"         /* QMF Exec Module       */
qmfgddm="QMF.330.DSQMAPE"          /* QMF GDDM Map          */

sysout="T"         /* Sysout Class for QMF temporaries      */

db2subs="DB2T"     /* DB2 Subsystem (empty=default)         */
db2aplan=""        /* DB2 QMF Access Plan (empty=default)   */

/*************************************************************/
```

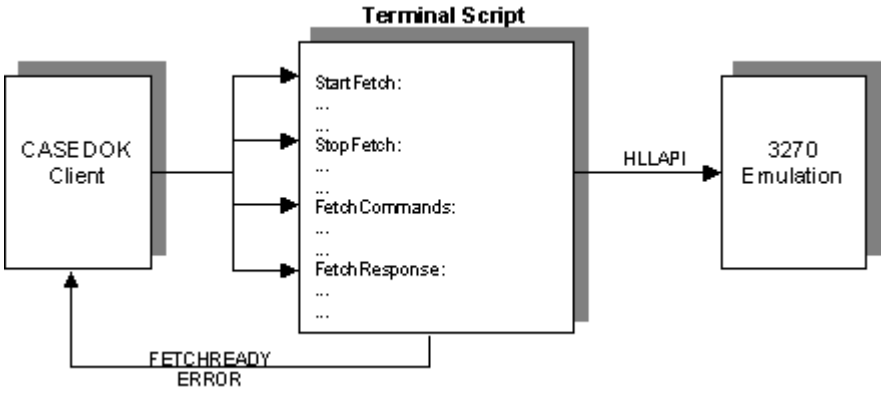
For batch you have to customize the server program "BATCH", which is a REXX procedure to run the FETCH server in batch. The following parameters (bold lines) have to be customized:

```
//USERnnnT JOB 0000000,'CASEDOK Fetch',
//         MSGCLASS=T,MSGLEVEL=(1,1)
//**
//** CONFIGURATION
//**
//** This batch needs the following customizations:
//** - An individual job header must be provided
//**   -> Fill in the jobname and the account
//**   -> Add site specific instructions
//** - The QMF installation parameters below must be adapted
//**   -> Adapt the dataset names to your installation
//** - The DB2 settings in the server commands at the end of
//**   this job must be changed if the defaults do not work
//**   -> Change 'set db2subs' to 'set db2subs XXXX'
//**   -> Change 'set db2aplan' to 'set db2aplan XXXXXX'
//**
//SETQL     SET QMFLIB=QMF330.DSQLOAD
//SETQC     SET QMFCLIST=QMF330.DSQCLSTE
//SETQE     SET QMFEXEC=QMF330.DSQEXECE
//SETQM     SET QMFGDDM=QMF330.DSQMAPE
//SETS      SET SYSOUT=T
//...
```

```
//*******************************
//** Terminal Output and Input **
//*******************************
//SYSTSPRT DD SYSOUT=&SYSOUT
//SYSTSIN  DD *
exec casedok(fetch) exec
checkpoint ----server started----
set queries CASEDOK.QUERY(*)
set reports CASEDOK.*.DATA
set confirm off
checkpoint ----DB2 settings:----
set db2subs
set db2aplan
dump
checkpoint ----processing commands----
service qmf start
output response
input commands
output
checkpoint ----done----
q
```

4.6.2.4 Customizing the terminal script(s)

A terminal script (*.TSC) is a program which controls the communication between the CASEDOK client and the CASEDOK FETCH server. The script program uses the HLLAPI interface to talk with the 3270 emulator program. A script is always triggered by the CASEDOK client through one of 4 possible states: StartFetch, StopFetch, FetchCommands, FetchResponse, and always ends with the FETCHREADY or ERROR state, or any other undefined state. FETCHREADY tells the CASEDOK client that the script transaction was successful. ERROR tells the CASEDOK client that the script found an unpredictable situation.



A script is an independant program with an own logic (state-event-machine). The scripting technique allows you to automate the communication between the CASEDOK client and the FETCH server. The script language provides different types of elements such as guards, actions and transitions, and also allows to work with variables. You can find a detailed description and a complete reference of the script commands in the FETCH.TSC script, which is used as standard script for seamless-forground mode. Further scripts (*.tsc) are provided with the product. Use these scripts as examples for your own scripts.

To customize the standard script for seamless mode, open the **FETCH.TSC** script and search for the word **'Customize':**

```
StartFetch:
  # ------------------------------------------------+
  # Assign some useful variables                    #
  # Connect to terminal session;                    #
  # Look for TSO prompt; start FETCH server;        #
  # Take screen shots in case StartFetch fails      #
  # ------------------------------------------------+
    Set(%args%,set chunk 15000)
    Set(%output%,)
    Set(%+%,%0D0A%)
    ############################################################
    ###### ==> Customize your 3270 emulation below <== ######
    Set(%session%,A)
    Set(%hllapidll%,PCSHLL32)
    Set(%transferoptions%,ASCII CRLF QUIET)
    ###### ==> Customize your 3270 emulation above <== ######
    ############################################################
    Connect(%hllapidll%,%session%)
  Match(%error%,.)
    ERROR in StartFetch
```

This part of the script shows the assignments of some script variables (SET) and the connection to a HLLAPI terminal session (CONNECT). If the connect action fails, the error variable contains the reason of the failure and the script ends in the undefined state 'ERROR in StartFetch'.

To customize the script to your site-specific environment, you must adapt the following parameters:

Set(%session%,A)	Assign the session ID (A, B, ...) to the script variable %session%.
Set(%hllapidll%,PCSHLL32)	Assign the name of the HLLAPI DLL to the script variable %hllapidll%. For example, 'PCSHLL32' is the DLL-name used by IBM's 3270 emulation program 'Personal Communications'.
Set(%transferoptions%,ASCII CRLF QUIET)	Assign the file transfer options ASCII CRLF QUIET to the script variable %transferoptions%.

You also need to define, how your TSO prompt looks like. So again search for the word 'Customize' to find 3 places where the TSO prompt is defined:

```
  ############################################################
  ###### ==> Customize additional prompts below <== ######
  See(READY,1,-1)
  See(ISPF Command Shell,25,-3)
  ###### ==> Customize additional prompts above <== ######
  ############################################################
```

See(READY,1,-1)	Looks for the literal 'READY' on the terminal screen at an offset of 1 column and -1 row away from the cursor.
See(ISPF Command Shell,25,-3)	Also looks for the literal 'ISPF Command Shell' on the terminal screen at an offset of 25 columns and -3 rows away from the cursor.

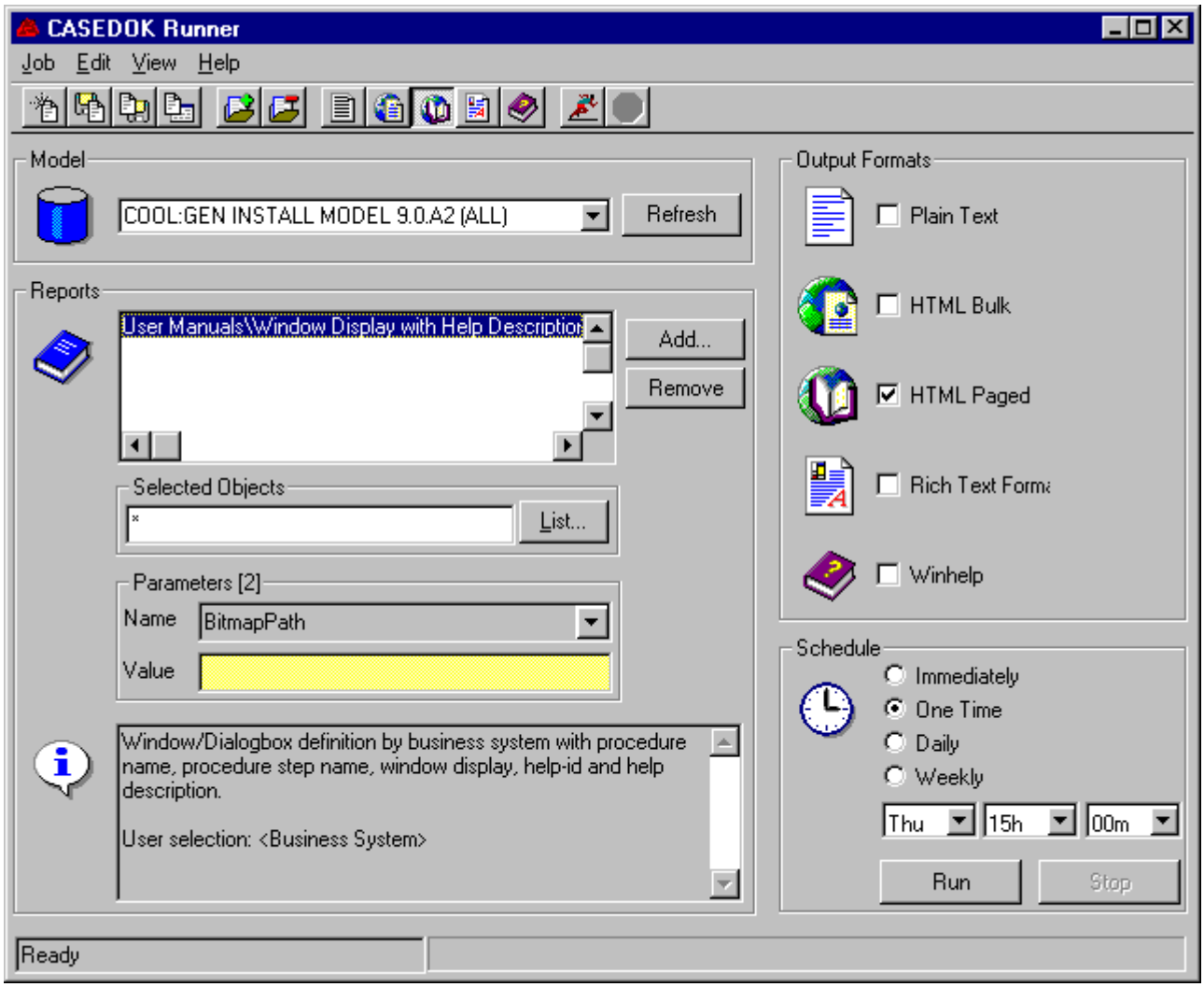
4.6.2.5 CE/MVS runtime parameters

The following runtime parameters are stored in the CASEDOK.INI file. To change these parameters, select **<View> <Settings>** from the menu bar of the CASEDOK Runner window.

Parameter	**Register**	**Description**
Macros	Reports	Location of the CASEDOK macrofile: <d:><path><filename>. The file contains the macros used in the CASEDOK reports. Macros are customizable text variables such as titles, object names or property values. For more information please read the description in the macrofile.
DocSuper	Reports	Document directory where generated documents are stored. This may be a local drive or a netdrive.
RepositoryInterface	Repository	Select your customized terminal script. This script must usually be adapted to the site-specific host environment (see chapter Customizing the terminal script(s).
SQLTABP	Repository	Table prefix for the CE database tables (database table owner). Note: Your TSO user ID is used for QMF/DB2 access and for checking your rights to see models.
BICOKOUT	Debugging	If a filename (<d:><path><filename>) is specified, extensive message logging is written to this file. Activate this logging option only for debugging purposes, because the performance will strongly be affected.
HSTBLKSIZE	Debugging	Limit for the number of objects expanded in a single step on the host. A larger value increases overall performance, but results in slow response times. Use a smaller value if you experience communication timeout problems. Defaults to 700.

5 How to run a report

Select <**RUNNER**> from the Windows Start menu to launch the application. From the CASEDOK RUNNER main window you can configure and start jobs. A job specifies: One model, one or more reports (with optional selections and parameters), one or more document formats and optional schedule settings. The job can be saved and reloaded for later runs.



Model selection:

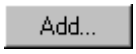
From the model list, select the COOL:Gen model on which you want to run a report. The model list is locally saved from one CASEDOK session to the next. Press the <Refresh> button to update the model list from the encyclopedia database.

CSE version: You may be prompted for the database userid and password to get access to the CSE repository.

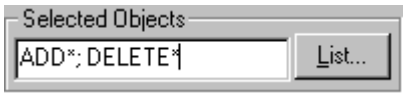
CE version: The procedure depends on the underlying terminal script you are using. For seamless mode, you must first prepare a TSO prompt in your emulator session, before pressing the <Refresh> button. **Attention:** Do not enter data or replace the cursor in the session window to not disturb the client-host communication!

Report selection:

Add as many reports to the job as you like. Press the [Add...] button to open a file selector and choose the desired report definition file. Look in directories to find reports to various themes. All reports in the job are generated within the same run.



Some reports are designed to depend on parameters. Select the parameter you want to change and enter the value in the field. See the info field for descriptions.



Many reports can be restricted to show only a part of the model. You can select the objects you wish to include in the report from a list, or you can enter a selection pattern. Press the [List...] button to show the list of objects in the current model to select from. When entering a pattern, you can use wildcards '*' and '?'. You can enter several patterns separated by a semicolon ';'.

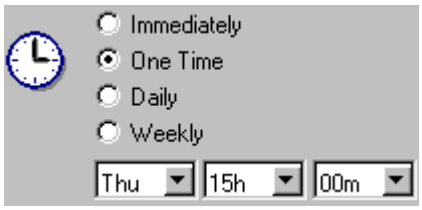
To inspect the objects in the current model, select **<View><Walker>** from the menu bar. This starts the CASEDOK WALKER tool with the focus from the current report of the job. For more details about WALKER see the **CASEDOK WALKER User's Guide**.

Format selection:

Select one or more text formats, in which the document is to be created. Available formats are:

***.TXT:** Plain Text
***.HTM:** HTML (bulk or paged) for Netscape or MS Internet Explorer. The paged format requires browser versions 4 or higer with Cascaded Style Sheet (*.CSS) support. The CASEDOK Style Sheet (Site.css) can be customized to site-specific needs.
***.RTF:** Rich Text Format for MS Word, WordPerfect, AmiPro, WordPro a.o.
***.HLP:** Windows Help (32bit)

Schedule settings:



You can schedule the creation of the reports for a later time or for periodical execution:

Immediately: Execution starts immediately when the job is started.
One Time: Execution starts at the specified day and time. You can enter a date within up to one week.
Daily: Execution starts every day at the given time.
Weekly: Execution starts every week at the specified day and time.

CASEDOK RUNNER needs to stay open while there is an active job waiting for scheduled execution.

Start job:

Starts the documentation job. All reports in the job list are created for all selected formats in the same job. **Attention for CE/MVS users:** Do not enter data or replace the cursor in the session window to not disturb the client-host communication!

When the reports are finished, a window listing the results appears. Double-click in the list to open a report. Click on the [Messages...] button to see messages.

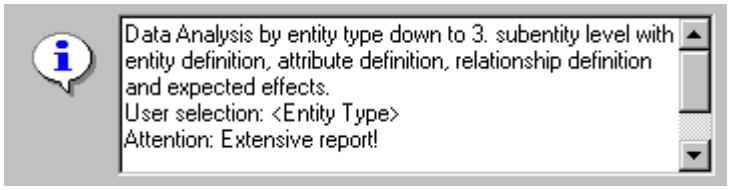
Stop job:

Stops the running job. CASEDOK RUNNER assures that partial documents are closed correctly and can be opened with the appropriate publisher or viewer.

Save/Load job:

The current job parameters can be saved in a job file and reloaded for a later run. The job file includes the following information: model, job report(s) (incl. data selection) and text format(s).

Report description:

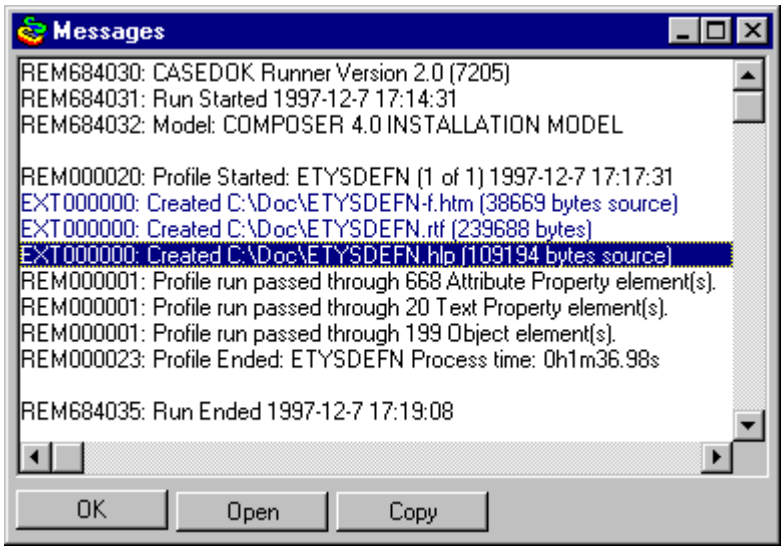


Status line:

Please look at the status line at the bottom of the main window to see information about parameter selection(s) or information about the current run.

5.1 Runner Messages

Messages window:



The 'messages' window shows some statistics about the latest run, such as document name, document size, number of processed objects and processing time. Double-click on a line with a document name opens this document in the related publisher or viewer. With <COPY> you can clip the contents of the 'messages' to the clipboard in order to paste it in a file. The statistics will be lost when the CASEDOK session ends. A closed 'messages' window can be reopened by selecting **<View><Messages>** from the main window menu.

6 Macrofiles

In all of the CASEDOK RUNNER standard reports, text variables (enclosed in []brackets) are used instead of fixed text strings. This concept allows you to redefine the default literals by your own terms, e.g. to adapt a report title or to translate COOL:Gen object names to French, or to substitute a property value.

The text variables are stored in a macrofile. The default macrofile delivered with CASEDOK RUNNER is 'ENGLISH.MCO'. You can create your own macrofile with your individual terms. Save the default macrofile as a new file (extension .mco is not important) and adapt the terms to your individual needs. Please read the comment inside the macrofile to understand the specific syntax.

If a variable is not resolved in the generated document, you will see the name of the variable in brackets ([variable]. Verify the <macros> parameter in the CASEDOK Configuration window to activate the right macrofile, or if the parameter is correct, the variable is missing and you must add it to the macrofile.

7 Troubleshooting

There can be many different reasons, why a CASEDOK RUNNER job is not successful. We do not try to list all potential problems and possible solutions. But we made a big effort to bring up significant error messages to the user interface. So in case of error, read carefully the information in the CASEDOK RUNNER 'Messages' window.

An additional way to trace a problem situation is to activate the logging facility (see <BICOKOUT> and <SQLTrace> parameters in the CASEDOK Configuration window).

7.1 Frequently Asked Questions

Q: The list of created reports is empty. In messages, it says 'no valid output streams'.

A: The report you would like to create is probably still opened in MS Word, so CASEDOK can't write to the file.

Q: CASEDOK goes down when I try to run a report.

A: This can be caused by an incompatibility in the workstation model access DLL with COOL:Gen 4.x. Upgrade to COOL:Gen 5.0 or later.

Q: The reports are empty and messages say 'EAPI_MODEL_INCOMPATIBLE_RC'.

A: You can only access DAT files which are not currently open in the Toolset. You can only access models of the current COOL:Gen release.

INDEX