

Software AG Mainframe Product Licensing

This document describes Software AG's mainframe product licensing procedures, utilities, and license file. The following topics are covered:

- Distributed License Libraries
 - License File
 - Transferring a License File from PC to a z/OS Host Using FTP
 - Transferring a License File from PC to a z/VSE Host Using FTP
 - Transferring a License File to a z/VM Host
 - Transferring a License File from PC to a BS2000 Host Using FTP
 - License File Installation
 - Product License Check FAQs
 - Using The License Utility: LICUTIL
-

Distributed License Libraries

All Software AG mainframe products are distributed with licensing libraries that contain the files necessary to license your Software AG mainframe software. This section describes these data sets. All licensing data sets have names starting with the characters "MLC".

z/OS Licensing Data Sets

The following licensing data sets are provided in z/OS environments:

Data Set Name	Description
MLCvrs.LOAD	Mainframe license check load library.
MLCvrs.JOBS	Mainframe license check example jobs.

z/VSE Licensing Data Sets

The following licensing data sets are provided in z/VSE environments:

Data Set Name	Description
MLCvrs.LIBR	Mainframe license check load library.
MLCvrs.LIBJ	Mainframe license check example jobs.

z/VM Licensing Data Sets

The following licensing data set is provided in z/VM environments:

Data Set Name	Description
MLC vr s.TAPE	Mainframe license check library.

Note:

Software AG licensing software is not provided for VM/GCS environments.

BS2000 Licensing Data Sets

The following licensing data sets are provided in BS2000 environments:

Data Set Name	Description
MLC vr s.MOD	Mainframe license check load library.
MLC vr s.JOBS	Mainframe license check example jobs.

License File

You must install a valid license file on all mainframe platforms in which your Software AG mainframe product is installed. Your individual product license is shipped on the installation tape. It can also be shipped by e-mail, if desired. License files have file names in the format "*pppvr.xml*", where *ppp* is the three-character product code of the Software AG product (for example, ADA for Adabas, NAT for Natural, or WCP for Entire Net-Work) and *vr* represent the *major and minor version numbers* (respectively) of the product.

Once the license file is received, you must use native FTP commands to transfer it to your mainframe host before using it during installation.



Warning:

Using utilities instead of native FTP commands for the license file transfer may corrupt the license key and thus prevent the later execution of your Software AG mainframe product. This applies, for example, to file transfer based on 3270 terminal emulations that do not provide a true binary file transfer, but convert specific characters.

The license file is provided as an XML document (encoding is US-ASCII). This document can be viewed using a browsing tool or text editor on a PC. It contains text, which represents the licensing information and a digital signature, the license key. Among other things, it displays Software AG legal notices and copyright information. The product license is comprised of environmental information.

Important:

Please note that the license file must remain in ASCII format -- even on the mainframe. It must not be modified. Any modification of the license file will invalidate the digital signature and the license check will fail. If the check fails, you will not be able to run the product. In the event of a check failure, please contact your Software AG technical support representative.

Transferring a License File from PC to a z/OS Host Using FTP

If a license file is supplied as an e-mail attachment, you must transfer the attached license file from the PC to the mainframe using native FTP commands.

**Warning:**

Using utilities instead of native FTP commands for the license file transfer may corrupt the license key and thus prevent the later execution of your Software AG mainframe product. This applies, for example, to file transfer based on 3270 terminal emulations that do not provide a true binary file transfer, but convert specific characters.

▶ **To transfer a license file from the PC to a mainframe z/OS host, perform the following steps:**

1. Save the product license file e-mail attachment on your PC's hard disk.
2. Open a command prompt window. In the command prompt window, change to the directory where you saved the license file.
3. Start an FTP session for communication with the z/OS host using the following FTP command:

```
ftp host-name
```

where *host-name* is the name of the z/OS host.

4. Enter your z/OS host login ID and password.
5. Once the FTP session has been established, you can optionally delete the default working prefix. To determine what the default working prefix is, enter the following command:

```
ftp>pwd
```

To delete the default working prefix, enter the following command:

```
ftp>cd ..
```

6. Specify the z/OS file system and data set prefix (for example, "ADABAS", "NATURAL", "NETWORK", or any other convenient prefix) for the license file. The following example sets the data set prefix to "ADABAS":

```
ftp>cd 'ADABAS'
```

7. Switch to binary data mode (the license file must retain its format as ASCII during the transfer):

```
ftp>binary
```

8. Specify that the data set for the license file must be written with RECFM=FB and LRECL=80.

```
ftp>quote site RECFM=FB LRECL=80 BLKSIZE=4000
```

RECFM=F is also supported for product license files.

9. Write the license as a data set on the z/OS system. For example, if the license file name is *ada81.xml*, you might enter:

```
ftp>put ada81.xml
```

This command will create a data set called "*prefix.ADA81.XML*", where *prefix* is the data set prefix you specified in Step 6. For example, if you requested a prefix of "ADABAS" in Step 6, the data set name would be "ADABAS.ADA81.XML". The data set will have RECFM=FB, LRECL=80, and the license information stored in the data set will be in ASCII format.

10. Stop your FTP session.

```
ftp>quit
```

Transferring a License File from PC to a z/VSE Host Using FTP

If a license file is supplied as an e-mail attachment, you must transfer the attached license file from the PC to the mainframe using native FTP commands.



Warning:

Using utilities instead of native FTP commands for the license file transfer may corrupt the license key and thus prevent the later execution of your Software AG mainframe product. This applies, for example, to file transfer based on 3270 terminal emulations that do not provide a true binary file transfer, but convert specific characters.

The FTP sample session described in these steps loads the license file onto a member of a z/VSE library. It is possible to load the license file onto a sequential z/VSE data set (with RECFM=F or FB and LRECL=80) instead. For more information, refer to your FTP documentation.

▶ **To transfer a license file from the PC to a mainframe z/VSE host, perform the following steps:**

1. Save the product license file e-mail attachment on your PC's hard disk.
2. Open a command prompt window. In the command prompt window, change to the directory where you saved the license file.
3. Start an FTP session for communication with the z/VSE host using the following FTP command:

```
ftp host-name
```

where *host-name* is the name of the z/VSE host.

4. Enter your z/VSE host login ID and password.
5. Once the FTP session has been established, you can optionally delete the default working prefix. To determine what the default working prefix is, enter the following command:

```
ftp>pwd
```

To delete the default working prefix, enter the following command:

```
ftp>cd ..
```

6. Point to the z/VSE library for the license file, (for example, "USERLIB"):

```
ftp>cd USERLIB
```

7. Point to the z/VSE sublibrary for the license file, (for example, "MYSUBLIB"):

```
ftp>cd MYSUBLIB
```

8. Switch to binary data mode (the license file must retain its format as ASCII during the transfer):

```
ftp>quote type I
```

9. Specify that the data set for the license file must be written with RECFM=FB and LRECL=80.

```
ftp>quote site recfm fb 80
```

RECFM=F is also supported for product license files.

10. Write the license as a member in the libraries on the z/VSE system you identified in Steps 6 and 7. For example, if the license file name is ada81.xml, you might enter:

```
ftp>put ada81.xml LICENSE.XML
```

This command will write the license file ada81.xml to a member called "LICENSE.XML". If you requested a library and sublibrary of "USERLIB" and "MYSUBLIB" in Steps 6 and 7, the LICENSE.XML file would reside in the USERLIB.MYSUBLIB library. The data set will have RECFM=FB, LRECL=80, and the license information stored in the member will be in ASCII format.

11. Stop your FTP session.

```
ftp>quit
```

Transferring a License File to a z/VM Host

If a license file is supplied as an e-mail attachment or on tape, you must transfer it to the mainframe. This section describes how to transfer the license file from tape and from e-mail (using FTP).

Note:

Software AG licensing software is not provided for VM/GCS environments.

► To transfer the product license file from tape to a source library, complete the following steps:

1. Refer to the *Report of Tape Creation* to accurately locate the license file on the tape.
2. If the tape is positioned somewhere other than the beginning of the tape, rewind it:

```
TAPE REW
```

3. Position the tape to the license file location, as shown in the *Report of Tape Creation*. For example, if the sequence number of ADAvrZVM file, as shown by the *Report of Tape Creation*, is *n*, you must position over 3 *n* - 2 tape marks (that is, FSF 1 for the first data set, FSF 4 for the second, etc.).
4. Run the following z/VM commands in sequence to transfer the file off of the tape.

```
FILEDEF IN TAP1 (RECFM FB LRECL 80 BLKSIZE 3120  
FILEDEF OUT DISK fn ft A (RECFM F LRECL 80 BLKSIZE 80  
MOVEFILE IN OUT
```

where *fn* is the file name of the license file and *ft* is the file type of the license file.

▶ To transfer the product license file e-mail attachment from Windows to a z/VM host, complete the following steps (make sure to switch to binary transfer and verify that the resulting data set has RECFM=F or FB and LRECL=80):

1. Save the product license file e-mail attachment on your PC's hard disk.
2. Open a command prompt window. In the command prompt window, change to the directory where you saved the license file.

```
cd dir
```

3. Start an FTP session with the z/VM host:

```
ftp host-name
```

where *host-name* is the name of the z/VM host.

4. Enter your FTP logon userid and password, as appropriate.
5. Switch the FTP data mode to binary:

```
binary
```

6. Request that the license data set to be written has RECFM=FB and LRECL=80:

```
quote site fix 80
```

7. Write the license file to the z/VM host. For example:

```
put fn.ft
```

where *fn* is the file name of the license file and *ft* is the file type of the license file.

The license file is written to your VM host.

8. Stop your FTP session:

```
quit
```

**Warning:**

Using utilities instead of native FTP commands for the license file transfer may corrupt the license key and thus prevent the later execution of your Software AG mainframe product. This applies, for example, to file transfer based on 3270 terminal emulations that do not provide a true binary file transfer, but convert specific characters.

Transferring a License File from PC to a BS2000 Host Using FTP

If a license file is supplied as an e-mail attachment, you must transfer the attached license file from the PC to the mainframe using native FTP commands.

**Warning:**

Using utilities instead of native FTP commands for the license file transfer may corrupt the license key and thus prevent the later execution of your Software AG mainframe product. This applies, for example, to file transfer based on 3270 terminal emulations that do not provide a true binary file transfer, but convert specific characters.

▶ To transfer a license file from the PC to a BS2000 host, perform the following steps:

Make sure to switch to binary transfer and verify that the resulting data set has file type SAM.

1. Save the product license file e-mail attachment on your PC's hard disk.
2. Open a command prompt window. In the command prompt window, change to the directory where you saved the license file.

```
cd dir
```

3. Start an FTP session for communication with the BS2000 host using the following FTP command:

```
ftp host-name
```

where *host-name* is the name of the BS2000 host.

4. Enter your FTP login ID and password.
5. Switch the FTP data mode to binary:

```
binary
```


6. Request that the license data set has a file type of SAM:

```
quote file *,fcctype=sam
```

7. Write the license file to the BS2000 host. For example, to write the *ada81.xml* license file, you would enter:

```
put ada81.xml
```

The license file is written to your BS2000 host.

8. Stop your FTP session:

```
quit
```

License File Installation

During the mainframe product installation process, the license file is loaded and processed as required by your Software AG mainframe product. Refer to the appropriate installation procedure for your Software AG mainframe product for further information.

Product License Check FAQs

1. **Why has Software AG introduced a license check for products on the mainframe?**
Software AG wants to ensure that customers run our software products only on mainframe machines for which they have valid and sufficient product license. This ensures that we have better control of the use of our software products. On other platforms (UNIX, Windows), the product licensing has been established for many years.
2. **What information is contained in a product license?**
A product license is a sequential file with US-ASCII text in XML format containing the following items:
 - Software AG header
 - customer information (Name, ID)
 - encrypted license key
 - license expiration date (or unlimited)
 - product information (product code, version, name)
 - environment information (operating system type, CPU ID, system name, capacity)

**Warning:**

Any modification of the license file will invalidate the digital signature and the license key check will fail. If the check fails, you will not be able to run the product. In the event of a check failure, please contact your Software AG technical support representative.

3. What exactly is meant by the CPU ID?

IBM defines the CPU ID as the "central processing complex node descriptor sequence number". It is the unique hexadecimal machine serial number without the machine model number.

z/OS

On z/OS systems, information about the CPU can be displayed on the operator console by entering the command:

```
D M=CPU
```

For example, suppose the console output from this command contains these lines:

```
EE174I 16.38.50 DISPLAY M 951
PROCESSOR STATUS
ID  CPU                SERIAL
00  +                  0FA10E2096
01  +                  0FA10E2096
CPC ND = 002096.S07.IBM.83.000000007A10E
CPC SI = 2096.V03.IBM.83.000000000007A10E
....
```

In this example, the CPU ID is "7A10E", as highlighted in the example above.

z/VSE

On z/VSE systems, information about the CPU can be displayed on the operator console by entering the operator command:

```
sir
```

For example, suppose the output from this command contains this line:

```
...
AR 0015 PROCESSOR = IBM 2096-V03 83 (7A10E83) LPAR = DAEX      No. = 0007
....
```

The CPU ID is highlighted in the example.

BS2000

On BS2000/OSD systems, the CPU ID (8-byte hexadecimal) consists of the machine serial number, the processor ID and the machine model number. The processor ID in the second byte of the CPU ID is ignored for the license check. Information about the CPU can be displayed by entering the following BS2000/OSD command:

```
/SHOW-SYSTEM-INFORMATION INFORMATION=*CPU-ID-LIST
```

For example, suppose the output from this command contains these lines:

```
...
%CONFIGURATION           = 7.500- S140-20A      «
%CPU-ID-LIST :   ADR     0   = 1D02301375000000 «
...
```

The most relevant information is shown in these lines. The first value identifies the machine type as "7.500- S140-20A", the second value identifies the physical CPU ID as "1D02301375" (trailing zeros can be omitted).

z/VM

On z/VM systems, the CPU ID can be determined entering the command:

```
q cpu
```

Suppose the output from this command is:

```
CPUID = FF07A10E20968000
```

The first byte will always be "FF", followed by the CPU-ID (highlighted in the example).

4. Which mainframe products use the Software AG common mainframe license routines?

- Adabas (version 8.1.4 or later)
- Com-plete (planned)
- Entire Net-Work (version 6.2.1 or later)
- Entire System Services (planned)
- Natural (planned)
- webMethods EntireX (planned)

5. How is the product license file installed?

The product license file is supplied on the individual customer installation tape or separately via an e-mail attachment. For information on its installation, refer to the installation documentation for your product. For information on transferring the license file from a PC to a z/OS, z/VSE, z/VM, or BS2000 host, read the various *Transferring a License File from PC to a xxx Host* sections .

In addition, steps for actually installing the license file are provided in the installation steps of your software product.

6. How can I read the product license file?

- The product license file can be read on a PC using an XML editor (file type *.xml*) or any PC-based text editor.

- The license utility, LICUTIL, provided with the Software AG mainframe license check software includes a function that allows you to display the license (Display License function). For more information about the license utility, read *The License Utility: LICUTIL*.
- When Natural is installed and runs, the license information can be displayed using function "L" of the Natural SYSTP utility.
- When Adabas, Com-plete, Entire Net-Work, Entire System Services, and webMethods EntireX are installed and run, they print a formatted license file into their logging files.

7. How can I display machine-specific data?

Use the license utility, LICUTIL, to display machine-specific data. For more information about the license utility, read *The License Utility: LICUTIL*.

8. When is the product license checked?

The product license is checked during every time the product is initialized. In addition, the product license is checked once a day.

Thirty days before the license expires, license check failure messages are produced. Your software product will still function, but these messages warn you that it is time to obtain a new license.

9. Which items of a product license are checked?

Product license items are divided into three logical license groups (License Key, Product-Specific Information, and Machine-Specific Information) and checked in that order. If an inconsistency in any one of these groups is detected by the license checker, the succeeding items of that logical license group are also checked. Then the check terminates with corresponding error messages. For example, if the license key expires, the license key check terminates with an error before the license checker checks any product-specific or machine-specific information.

The following table identifies the product license items comprising each logical group:

Checking Order	Logical License Group Name	Product License Items Included
1	License Key	The encrypted license key
2	Product-Specific Information	<ul style="list-style-type: none"> ● The expiration date (if any) ● The operating system (z/OS, z/VSE, z/VM, or BS2000/OSD) ● The product code (for example, ADA, COM, EXX, NAT, or WCP) ● The product version, in either "v.r" or "v.r.s" format. For further information on product versions, refer to the description of "v.r" or "v.r.s" in the product documentation Glossary.
3	Machine-Specific Information	<ul style="list-style-type: none"> ● The machine CPU ID ● The machine capacity (only in z/OS, z/VSE, and z/VM environments) ● The logical partition

10. Does the Software AG mainframe license checker software support a disaster recovery scenario?

Yes. Please contact your Software AG sales representative to obtain a license file which is able to support disaster recovery scenarios. The parameter DisasterRecoveryCPUIDs in this license file allows you to specify one or more CPU IDs of the machines to be used when your production machine fails. If you do not have these CPU IDs available, it is possible to specify the value "Unknown". In this case, the license checker issues a warning message when a disaster recovery scenario is assumed.

In a disaster recover scenario, no machine data besides the CPU ID is checked.

11. What happens if the product license expires or is incorrect, insufficient, or not installed?

The reaction varies by Software AG mainframe product. For more information, refer to the documentation for the product.

Thirty days before the license expires, license check failure messages are produced. Your software product will still function, but these messages warn you that it is time to obtain a new license.

12. How can I get a new product license file if the delivered license file is insufficient for my environment?

Contact your Software AG sales representative to get a correct product license file according to your contract. It can be sent by e-mail either in ASCII format or in the converted assembler input format.

13. Do I need separate product licenses for different machines?

No, it is possible to have multiple CPU IDs defined in one license file.

Using The License Utility: LICUTIL

The license utility, LICUTIL, is a utility provided with Software AG's mainframe license checking software. It can perform the following functions:

- Check the license file. LICUTIL is mainly intended to be used to check the license file during the installation of the product. The original license file must be provided as input along with the product code and version. If the license file is not valid, appropriate error messages are returned.
- Display the license file and machine data. LICUTIL is able to display the contents of the license file and machine data, such as the CPU ID, LPAR, or MSU information.
- Convert the license file into an assembler source module. LICUTIL can be used to convert the license file into an assembler source module that can then be used to assemble and link a license module. For more information about using license key modules, refer to the documentation provided with your Software AG product.

The following table briefly describes the functions and parameters in the license utility. More detailed information of each function is provided later in this section:

Function	Parameters	Description
CHECK	---	Check the product license.
DISPLAY	ALL	Display the product license items and machine data.
	LICENSE	Display only the product license items.
	MACHINE	Display only the machine data.
END EXIT	---	Stop the license utility.
HELP	---	Produce a list of available LICUTIL functions and the syntax of the functions.
MAKE	---	Create an assembler source module from the license file.
SET	DATE=yyyy-mm-dd	Set the date for a license check. The default is today.
	OR	
	DATE= yyyymmdd	
	FORMAT= {FREE LEFT [STRUCTURED]}	Produce output without XML tags (FREE), with each XML tag in a new line (LEFT), or with each XML tag indented (STRUCTURED). The default is FREE.
	PRODVER=pppv.r.s	Set the product code and version, where <i>ppp</i> is the Software AG 3-character product code and <i>v.r.s</i> is the three-digit version number.

Error messages for the license utility all begin with the prefix MLC. They are documented with the rest of your Software AG product messages.

This section covers the following topics:

- CHECK Function: Check the Product License
- DISPLAY Function: Display License Items and Machine Data

- END/EXIT Functions: Terminate the License Utility
- HELP Function: View LICUTIL Help
- MAKE Function: Convert the License File to Assembler Source
- SET Function: Set LICUTIL Options
- JCL Examples and Requirements

CHECK Function: Check the Product License

The CHECK function can be used to check the license file against the product's execution environment. There are no parameters, however to run the CHECK function successfully, you must have previously specified the SET PRODVER function to provide the product code and version number.

The following items are checked, in the following order:

1. License key
2. Product-specific data, including the product code, version, operating system, and expiration date.
3. Machine-specific data, including the CPU IDs, the capacity (on only z/OS, z/VSE and z/VM systems), and the logical partition.

The syntax of the CHECK function is:



Sample JCL or JCS is provided for the CHECK function in member LICCHECK of the Software AG mainframe license JOBS data set (MLCvrs.JOBS).

Example:

In the following example, the product code and version are set to ADA8.1.4 (Adabas 8.1.4) and then the product license is checked.

```
SET PRODVER=ADA8.1.4  
CHECK
```

DISPLAY Function: Display License Items and Machine Data

Use the DISPLAY function to display the contents of the product license items (license file, product-specific information, and machine-specific information) and to display machine data (CPU ID, machine type, model ID, LPAR, MSU) about the machine on which the utility is being run.

The syntax of the DISPLAY function is:


```
DISPLAY { ALL | LICENSE | MACHINE }
```

The parameters are described in the following table. At least one parameter is required.

Parameters	Description
ALL	Display the product license items and machine data.
LICENSE	Display only the product license items.
MACHINE	Display only the machine data.

Sample JCL or JCS is provided for the DISPLAY function in member LICMDATA of the Software AG mainframe license JOBS data set (MLCvrs.JOBS).

Examples

In the following example, both license and machine data is displayed:

```
DISPLAY ALL
```

In the following example, only license data is displayed:

```
DISPLAY LICENSE
```

In the following example, the license data is displayed in structured format:

```
SET FORMAT=STRUCTURED
DISPLAY LICENSE
```

In the following example, only machine data is displayed:

```
DISPLAY MACHINE
```

END/EXIT Functions: Terminate the License Utility

Use the END or EXIT functions to terminate the license utility. These functions should be used if LICUTIL is running interactive in environments other than batch (for example TSO under z/OS).

The syntax of the END and EXIT functions is:

```
{ END | EXIT }
```

HELP Function: View LICUTIL Help

Use the HELP function to review a list of the available LICUTIL functions and their syntax.

The syntax of the HELP function is:

```
HELP
```

Sample Output

Here is some sample output from this function:

```
>>>HELP
LICUTIL reads a Software AG product license from SYSUT1.
It has the following functions:
  CHECK   : Check license read from SYSUT1
  DISPLAY : Display license and machine
  END     : Terminate LICUTIL
  MAKE    : write license for assembly to SYSUT2
  SET     : sets execution defaults

SET keyword=value accepts the following keyword(s):
  DATE=YYYY-MM-DD or YYYYMMDD - Set date for license check (default TODAY)
  PRODVER=zzzv.r.s             - Set product code zzz and version release
                               SM-level v.r.s for license check
  FORMAT=LEFT/STRUCTURED/FREE - Set format of license display
```

MAKE Function: Convert the License File to Assembler Source

The MAKE function can be used to convert the license file into an assembler source. After assembling this source (and linking it), the resulting license module can be used by defined Software AG products (for example Adabas or Natural) instead of the license files. For more information, please read the documentation for your specific Software AG mainframe product.

The syntax of the MAKE function is:

```
MAKE
```

Sample JCL or JCS is provided for the CHECK function in member LICMAKE of the Software AG mainframe license JOBS data set (MLCvrs.JOBS).

Sample Output

Here is some sample output from this function when run on a z/OS system:

```
MLC9006 License converted to assembler output. 042 records written
to SYSUT2.
```

SET Function: Set LICUTIL Options

Use the SET function to set options for the license utility.

The syntax of the SET function is:

```
SET [ DATE= { yyyy-mm-dd | yyyymondd } ]
    [ FORMAT = { FREE | LEFT | STRUCTURED } ]
    [ PRODVER = pppv.r.s ]
```

The parameters are described in the following table. At least one parameter is required.

Parameters	Description
DATE	Sets the date for the license check. The default is today. The format of the date can be either <i>yyyy-mm-dd</i> or <i>yyyymondd</i> , where <i>yyyy</i> is the four-character year, <i>mm</i> is the two-digit month, <i>mon</i> is the three-character month abbreviation, and <i>dd</i> is the two-digit day.
FORMAT	Sets the display format for the DISPLAY function. There are three different formats: <ul style="list-style-type: none"> ● Specify FREE (the default) to produce output with XML tags. ● Specify LEFT to produce output with XML tags ● Specify STRUCTURED to produce output with each XML tag indented.
PRODVER	Sets the product code and version for the CHECK function. The format in which the product code and version should be specified is <i>pppv.r.s</i> , where <i>ppp</i> is the three-character Software AG product code (such as ADA, NAT or WCP) and <i>v.r.s</i> is the three-digit version number.

Sample JCL or JCS is provided for the DISPLAY function in member LICCHECK of the Software AG mainframe license JOBS data set (MLC*vrs*.JOBS).

Example

In the following example, the format of the DISPLAY output will not include XML tags and the product and version are set as Adabas 8.1.4:

```
SET FORMAT=FREE
SET PRODVER=ADA8.1.4
```

JCL Examples and Requirements

Sample JCL of the license utility functions can be found in the following members of the Software AG mainframe license example libraries (MLCvrs.JOBS in z/OS and BS2000 environments, MLCvrs.LIBJ in z/VSE environments, and MLCvrs.TAPE in VM environments).

Member	Functions Included	Demonstrates how to...
LICCHECK	SET PRODVER=PPPv.r.s DISPLAY CHECK	Check a license file with LICUTIL.
LICMAKE	MAKE	Convert the license file into an assembler source.
LICMDATA	DISPLAY MACHINE	Display machine data. This function can be used to send Software AG the required data for a license file.

This section covers the following topics:

- z/OS and z/VM JCL Requirements
- z/VSE JCL Requirements
- BS2000/OSD JCL Requirements

z/OS and z/VM JCL Requirements

The following file assignments are used by the license utility in z/OS and z/VM environments:

DD Name	Description
SYSIN	Input of LICUTIL control parameters.
SYSPRINT	Output of LICUTIL run.
SYSUT1	Input of original license file.
SYSUT2	Output of license converted to assembler source.

The following table shows which file assignments are required in the JCL by each LICUTIL function (an X in a table cell indicates that the file assignment is required; a blank cell indicates that file assignment is not required):

License Utility Function	DD Name			
	SYSIN	SYSPRINT	SYSUT1	SYSUT2
CHECK	X	X	X	
DISPLAY ALL	X	X	X	
DISPLAY LICENSE	X	X	X	
DISPLAY MACHINE	X	X		
MAKE	X	X	X	X
HELP	X	X		

z/VSE JCL Requirements

The following file assignments are used by the license utility in z/VSE environments:

File Assignment	Description
SYSIPT	Input of LICUTIL control parameters and the input of original license file. Note: The license file has to be stored in a sublibrary.
SYSLIST	Output of LICUTIL run.
SYSPCH	Output of license converted to assembler source.

The following table shows which file assignments are required in the JCL by each LICUTIL function (an X in a table cell indicates that the file assignment is required; a blank cell indicates that file assignment is not required):

License Utility Function	File Assignments		
	SYSIPT	SYSLIST	SYSPCH
CHECK	X	X	
DISPLAY ALL	X	X	
DISPLAY LICENSE	X	X	
DISPLAY MACHINE	X	X	
MAKE	X	X	X
HELP	X	X	

BS2000/OSD JCL Requirements

The following file assignments are used by the license utility in BS2000 environments:

System File	Description
SYSDTA	Input of LICUTIL control parameters.
SYSOUT	Output of LICUTIL run.

Link Name	Description
SYSUT1	Input of original license file.
SYSUT2	Output of license converted to assembler source.

The following table shows which file assignments are required in the JCL by each LICUTIL function (an X in a table cell indicates that the file assignment is required; a blank cell indicates that file assignment is not required):

License Utility Function	File Assignment			
	SYSDTA	SYSOUT	SYSUT1	SYSUT2
CHECK	X	X	X	
DISPLAY ALL	X	X	X	
DISPLAY LICENSE	X	X	X	
DISPLAY MACHINE	X	X		
MAKE	X	X	X	X
HELP	X	X		