SERENA

SERENA®
SER10TY™ 4.3

User's Guide

Serena Proprietary and Confidential Information

Trademarks

U.S. Government Rights

Publication date: April 2016

Table of Contents

Welcome to Serena SER10TY

This document describes Serena® SER10TY™ (Mainframe License Manager), a product of Serena Software, Inc.

SER10TY is used for licensing Serena mainframe products and for validating licenses.

Audience and Scope

This document is intended for experienced system administrators with skills in installation, security, and configuration.

The purpose of this manual is to:

- Describe the licensing process for Serena mainframe products.

- Provide instructions for applying and updating licenses.

Manual Organization

This manual is organized as follows:

This chapter ...	Contains this information ...
1	Introduction. Gives an overview of the licensing process.
2	System and Application Requirements.
3	Licensing Procedures.
4	License Test Utility. Explains how to test your licenses.
Appendix A	License Data. Shows a sample license and describes the elements.

How to Use This Book

This book is intended to be used in sequential order for chapters 1 through 4.

Related Documents

The installation guide for your Serena product contains information on how to install SER10TY components on the mainframe.

Following is a list of installation guides:

Serena® ChangeMan® SSM Installation Guide

Serena® ChangeMan® ZDD Server Installation Guide

Serena® ChangeMan® ZMF Installation Guide

Serena® Comparex® for z/OS Installation and Setup

Serena® StarTool® DA Installation Guide

Serena® StarTool® FDM Installation Guide

Serena® StarTool® IOO Installation Guide

Chapter 1

Introduction

This chapter describes the process used for licensing Serena mainframe products and options.

Compatibility

You may have existing Serena Mainframe products that use an earlier version of SER10TY (2.4.4 and lower). There is no problem running products using different versions of SER10TY. The products using version 2.4.4 and lower will continue to validate licenses using the SERVERAO module; whereas the products using SER10TY 3.1 and later will validate licenses using the new process described in this chapter.

Licensing Process

The following illustration gives a high-level view of the licensing process:
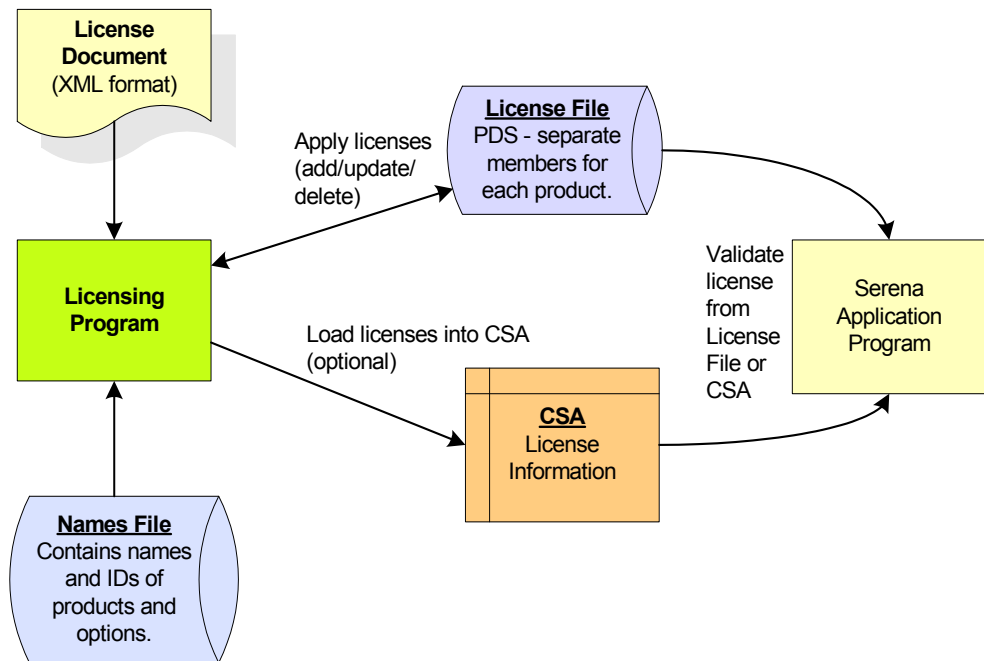
The licensing program performs two basic functions:

- Converts an XML-formatted license document to binary format and stores it in the license file.

- Loads licenses from the license file into common storage (optional).

Serena application programs validate licenses by accessing either the license file or common storage (CSA).

The following topics describe the files and processes:

-

-

-

-

Files Used

License Document

The XML-formatted license document you received (as an email attachment) for your Serena products contains the information you need to activate the base products and their options. A single license can cover multiple products. See Appendix A, "License Data" on page 35 for element descriptions and a license example.

License File

The license file is a partitioned data set with separate members for each product. A single license file can be used for multiple systems and products; separate license files can be used for different systems.

> **IMPORTANT!** If you have multiple Serena products using XML-format licenses, it is recommended that you use one license file and place it on shared DASD for all Serena products to access from any system. Using separate license files may cause an abend.

Names File

The names file is used to map product IDs to product names and option IDs to option names. Application programs reference this file to obtain the correct product name for messages.

The names file is provided in *somnode*.SERCOMC.V*n*R*n*M*n*.XML(NAMES). This file is updated whenever products or options are added or renamed.

The licensing program can be run with an out-of-date or incomplete names file; however, program messages and generated messages may have missing or incorrect product/option names.

Applying Licenses

Licenses are applied by running a batch job, using JCL provided in the software installer. Refer to your product's Installation Guide for the name of the JCL library; it is usually called

> *somnode*.SERCOMC.V*n*R*n*M*n*.CNTL(LICUPDAT)

You must modify the sample JCL to meet your site's requirements.

The licensing program reads the license document and adds/replaces the license in the license file. The parameters for applying new licenses are specified using the PARM parameter on the JCL EXEC statement, as follows:

- **ADD** merges new licenses into the existing licenses. Existing licenses are not changed; the new license information is appended onto the existing license. Also, expired licenses are removed from the file after the grace period has elapsed.

- **REP** replaces all existing licenses for products in the input file with the new licenses.

The following are recommendations for using ADD or REP:

- Use ADD if you already have the product licensed and the new licenses are just for adding a new option or CPU ID.

- Use REP if the new licenses are a complete set of licenses for the product.

Regardless of whether you use ADD or REP, a product's licenses are not updated if there are no licenses for that product in the input file.

The following members are created/updated for each product ("*nnnnn*" is a unique number assigned to each product):

LIC*nnnnn*	Binary license data, used by application programs for validating the license.
XML*nnnnn*	XML formatted license, for reference only.
BAK*nnnnn*	Backup of previous version of XML formatted license.

The licensing program stores the binary licenses in the LIC*nnnnn* members and places an updated XML version of the licenses in the XML*nnnnn* members. Before updating the licenses, a copy of the previous XML*nnnnn* member is saved as BAK*nnnnn*.

If the member name already exists, that member is updated; otherwise, a new member is created.

Licenses can be updated at any time by running the licensing program. Only the products and options specified in the license input are updated; other product and option information in the license file remains as is.

Loading Licenses into Common Storage (Optional)

Licenses can optionally be loaded into common storage (CSA). If you do not load your licenses into CSA, your applications will access the license file for validation. For more information on the different methods of validating licenses, and the pros and cons of each, see "Validating Licenses" on page 12.

Licenses are loaded into common storage by running the licensing program with the **LOAD** parameter; licenses for all products in the license file are loaded.

Applying new licenses with the ADD or REP parameter does not automatically load the new licenses into CSA; you must run the licensing program a second time with the LOAD parameter to update the licenses cached in CSA.

You can restrict which users can load licenses into CSA with a security profile. See "Security" on page 16 for details.

If licenses are loaded into CSA, the licensing program must be run after each system IPL to reload the licenses into CSA. The licensing program can be run at any time to update the licenses in CSA.

Validating Licenses

Serena application programs validate licenses by calling a license-check routine. The license checker accesses licenses by looking in the license file or in CSA.

Licenses can be loaded into CSA or the license file can be accessed in either of the following ways:

- Assembling the name of the file into the LICDSN load module that resides in the system link list or in the JOBLIB or STEPLIB of the product JCL.

- Using the SERLIC DD statement in the product JCL.

Validation Methods

The three methods available for license validation can be summarized as follows:

- LICDSN load module pointing to license file

- SERLIC DD statement pointing to license file

- Licenses loaded into CSA

Instructions on how to implement the different methods can be found in Chapter 3, "Licensing Procedures" on page 19. The specific topics are:

- "Create a LICDSN Module" on page 23 - provides instructions on how to create the LICDSN module for dynamically allocating the license file.

- "Add a SERLIC DD Statement to Your JCL" on page 24 - provides instructions on how to allocate the license file using a SERLIC DD statement.

- "Load Licenses into Common Storage" on page 24 - provides instructions on how to load and access licenses in CSA.

Order of Search

The license checker searches for the licenses in the following order:

1   If there is a SERLIC DD statement in the application JCL, the license checker tries to read the license file named on this DD statement.

2   If the SERLIC DD statement is *not* present, the license checker looks for the LICDSN module in the system link list or in the product STEPLIB or JOBLIB.

3   If the LICDSN module is not found, or it is the default module delivered with the product, then the license checker looks in CSA.

Pros and Cons of the Different Methods

LICDSN Load Module Pointing to License File	
Advantages	No changes to current JCL.
Disadvantages	Requires updating a module in the product load library.

SERLIC DD Statement Pointing to License File	
Advantages	Easy for installer or administrator to change when testing new Serena products or new versions of licensed products. Easy to see identity of license file. JCL checking utilities (JCL/PRO or ASG-JCLPREP) can guarantee production name for license file.
Disadvantages	Changes required to current JCL

Licenses Loaded into CSA	
Advantages	Universally available to Serena products on a z/OS image. No change to current JCL.
Disadvantages	Must go through systems programming department to get updated. Requires a job to be submitted to reload licenses after an IPL.

Messages

The message identifiers generated by the license-check routine are `LIC0000` through `LIC0099`, and `LIC9999`; the message text is self-explanatory. Messages are displayed as follows:

- For batch jobs and started tasks, messages are logged to the JESMSGLG output.

- For TSO users, messages are displayed at the terminal.

- Some applications may also echo messages to a SYSPRINT DD statement, or log them elsewhere.

Chapter 2

Requirements

System Requirements

Link List

If the Language Environment (LE) runtime libraries are *not* included in the link list, they will need to be added to the STEPLIB concatenation of every application that performs license checking.

The names of the LE runtime libraries as supplied by IBM are CEE.SCEERUN and CEE.SCEERUN2. These dataset names may have been changed at your site to meet local naming standards.

APF List

The load library containing the licensing load module (LICUPDAT), as well as the Language Environment (LE) runtime libraries, must be APF-authorized under either of the following conditions:

- Licenses are loaded into common storage. LICUPDAT must run APF-authorized in order to perform the load function.

- License checking is performed from any APF-authorized application, such as the SERNET started task.

The names of the LE runtime libraries as supplied by IBM are CEE.SCEERUN and CEE.SCEERUN2. These dataset names may have been changed at your site to meet local naming standards.

Security

You can control which users are permitted to load licenses into CSA. The load function of the licensing program (LICUPDAT) can be protected with a security profile as shown below.

Class:	FACILITY
Profile Name:	LICENSE.LOAD
Access Required:	READ

The commands to set up this profile for RACF are:

```
RDEFINE FACILITY (LICENSE.LOAD) UACC(NONE) OWNER(owner)
PERMIT LICENSE.LOAD CLASS(FACILITY) ACCESS(READ) ID(userid1 userid2 ...)
```

Application Requirements

Load Libraries

All applications that perform license checking must have the Language Environment runtime libraries available. If CEE.SCEERUN and CEE.SCEERUN2 are not part of the system link list, they must be included in the application STEPLIB concatenation.

Chapter 3
Licensing Procedures

This chapter provides information to assist you in applying licenses.

Required Components

If you have previously unloaded the licensing components from the software installer, you can skip this section and go to the next section, "Procedures".

SER10TY components are distributed for each Serena product on that product's software installer. The SER10TY components exist in data sets that also contain other Serena components.

The data set names for the SER10TY components start with

> SERCOMC.V*n*R*n*M*n*

where *n* is the Version, Release, and Modification level.

An example of a data set name would be SERCOMC.V8R1M1.XML.

Before continuing to the next section, unload the SER10TY data sets to their respective product libraries as defined in the product's installation guide.

Procedures

Follow the steps described in this section to apply your licenses and make them available to your Serena products:

Step 1: Allocate the License File

Step 2: Update Licenses

Step 3: Make Licenses Available

Step 4: Verify Licenses (Optional)

Step 1: Allocate the License File

If you already have a license file, skip this step and go to .

IMPORTANT! If you have multiple Serena products using XML-format licenses, it is recommended that you use one license file and place it on shared DASD for all Serena products to access from any system. Using separate license files may cause an abend.

The license file has the following attributes:

> RECFM: VB
>
> LRECL: 259 - 32760

A record length of 259 is recommended.

The JCL for allocating the license file can be found in:

> *somnode*.SERCOMC.V*n*R*n*M*n*.CNTL(LICALLOC)

Before running the job, specify the license file name and change the attributes if needed.

Step 2: Update Licenses

The JCL for adding and updating licenses can be found in:

> *somnode*.SERCOMC.V*n*R*n*M*n*.CNTL(LICUPDAT)

1 Specify one of the following values using the PARM parameter on the JCL EXEC statement:

Value	Description
ADD	Merges new licenses into the existing licenses. Existing licenses are not changed; the new license information is appended onto the existing license. Also, expired licenses are removed from the file after the grace period has elapsed.
REP	Replaces all existing licenses for products in the input file with the new licenses.

If no parameter is specified, the default is ADD.

Refer back to "Applying Licenses" on page 10 for more information about using ADD or REP.

2 Specify the data set names as follows:

DD Name	Description
STEPLIB	Specify the name of the SERCOMC load library as the first library. The Language Environment runtime libraries, CEE.SCEERUN and CEE.SCEERUN2 must be specified if they are not included in the system link list.
SYSLIB	Specify the name of your license file.
SYSNAME	Specify the name of your names file.
SYSIN	This is the XML-formatted license document you received as an email attachment. You can cut and paste it into the JCL or copy it into a separate file.

3 Run the job. Detailed program messages are written to the SYSPRINT DD.

A license report is available in SYSOUT. The following sample report was produced by running LICUPDAT with PARM='REP'. The input license data for products 204 and 207 replaced the existing license data on the license file.

```
PRODUCT  ID: 204   Name: ChangeMan ZDD    License type: Demo
OPTION   ID: 0     Name: Base            Expiration: 2010/11/21
-------------------------------------------------------------------
PRODUCT  ID: 207   Name: ChangeMan ZMF    License type: Demo
OPTION   ID: 0     Name: Base            Expiration: 2010/11/21
         ID: 1     Name: DB2             Expiration: 2010/11/21
         ID: 8     Name: ERO             Expiration: 2010/11/21
```

Step 3: Make Licenses Available

You may use any or all of the methods described in this section to make the licenses available to your Serena products. The instructions for each method are described in the following topics:

- "Create a LICDSN Module" on page 23
- "Add a SERLIC DD Statement to Your JCL" on page 24
- "Load Licenses into Common Storage" on page 24

> **IMPORTANT!** It is important to understand how license validation works before proceeding with this step. Refer back to "Validating Licenses" on page 12 for an explanation of the different methods, the pros and cons of each method, and the search order used by the application program when validating licenses.

Create a LICDSN Module

A default LICDSN module is provided in:

somnode.SERCOMC.V*n*R*n*M*n*.LOAD

You can update this LICDSN module using the JCL in:

somnode.SERCOMC.V*n*R*n*M*n*.CNTL(LICDSN)

1 Specify the following data set names:

- In the SYSIN DD data of the ASM step, specify the name of your license file.
- In the SYSLMOD DD of the LINK step, specify the name of your product load library.

2 Run the job.

3 Make sure the library specified in SYSLMOD is in the system link list or in the JOBLIB or STEPLIB of the product JCL.

> **NOTE** If there is a SERLIC DD statement coded in the application JCL, it will take precedence over the license file named in the LICDSN module. Otherwise, the license file named in the LICDSN module takes precedence over the licenses in CSA.

Add a SERLIC DD Statement to Your JCL

If your Serena product runs as an application under a SERNET started task, you can code the SERLIC DD statement in the started task JCL. Otherwise, you will have to code the SERLIC DD statement in all of the application JCL.

The SERLIC DD is allocated by specifying the data set name, without any member name, as follows:

```
//SERLIC DD DSN=somnode.LICENSE.DATA,DISP=SHR
```

> **NOTE** The license file coded on the SERLIC DD takes precedence over both the license file named in the LICDSN module and the licenses in CSA.

Load Licenses into Common Storage

When you applied your licenses with the ADD or REP parameter, they were not automatically loaded into CSA; you must run LICUPDAT a second time with the LOAD parameter to update the licenses cached in CSA.

> **NOTE** Licenses in CSA are the last order of search during license validation. Therefore, if there is a SERLIC DD statement coded in the application JCL, or a license file is named in the LICDSN module, licenses in CSA will not be accessed.

The LICLOAD JCL member loads licenses for all products in the license file into CSA. This JCL can be found in:

```
somnode.SERCOMC.VnRnMn.CNTL(LICLOAD)
```

1 Specify PARM='LOAD' on the JCL EXEC statement.

2 Specify the data set names as follows:

DD Name	Description
STEPLIB	Specify the name of the SERCOMC load library as the first library.
	The Language Environment runtime libraries, CEE.SCEERUN and CEE.SCEERUN2 must be specified if they are not included in the system link list.
SYSLIB	Specify the name of your license file.
SYSNAME	Specify the name of your names file.

3 Run the job. Program messages are written to the SYSPRINT DD. A license report is available in SYSOUT.

Step 4: Verify Licenses (Optional)

Your licenses should now be available. You can use the LICTEST utility to validate your licenses before using your Serena product. See Chapter 4, "License Test Utility" on page 29.

Return Codes

Return codes for the licensing program (LICUPDAT) are as follows:

Return Code	Explanation
0	The step completed successfully.
4	The step completed with warning messages.
8	When attempting to load licenses into CSA, no licenses were found in the license file.
12	There was an error in one or more of the licenses, and the step did not run to completion.

Messages

Messages generated by the licensing program are logged to the JESMSGLG output and echoed to the SYSPRINT DD. The message identifiers are `LIC0100` through `LIC0199`, and `LIC9999`; the message text is self-explanatory.

Troubleshooting

This section describes problems you may encounter and how to solve them.

Invalid Value for Key

If you receive the following messages:

```
LIC0161E Invalid value for "Key" element: (Line nn, Column nn).
LIC0102E License utility terminated with errors. Return code 12.
```

it is probably because you modified the license data, which caused the Key element to become invalid.

To correct this error, rerun the LICUPDAT program with the unmodified license as you received it from Serena.

No Applications with Valid Licenses or No License Found

With the XML licensing format, there is a change to the SERNET started task. The *apl* keyword, formerly used to assign a TCP/IP port to an instance running under the task, now also has the function of starting the specified instance. For example, one of the values of the *apl* keyword is ZDD. Coding "ZDD=*portid*" (or "ZDD" alone tells the SERNET started task that you want to run ChangeMan ZDD under this SERNET instance.

This new function of the SERNET *apl* keyword applies to all products that run as an instance under SERNET. If your product install guide contains a section on "SERNET Keyword Options", you can see all possible values of the *apl* keyword.

With this new on/off switch you may receive messages similar to the following:

```
SERxxxxE No applications with valid licenses; Shutting down
```

This message may indicate that there is no *apl* keyword in the started task. To correct this problem, add the keyword with the correct value to the SERNET parms. For example, to start an instance of ChangeMan SSM, add "SYN" to the parm list, as follows:

```
//SERSTC EXEC PGM=SERVER,
// REGION=0M,
// PARM='SYN,...'
```

If this keyword is in the started task and is correct, then the message indicates that the license is not valid.

You may also receive messages similar to the following (instead of or in addition to the above messages):

```
LIC0020E License file could not be opened: DD:SERLIC(LIC00204)
LIC0020E Trying to open non-existent file for READ
LIC0024E No license found for ChangeMan ZDD (Product ID 204)
LIC0043E ChangeMan ZDD terminating due to license validation errors.
SER3801I Serena/Network - Initialization in progress; CSA=00D000
SER3805I Execution Parameters:
SER3805I DDNAME=SERSYSIN
SER3805I SUBSYS=3
SER3805I CMN=portid
SER3805I ZDD=portid
SER3805I STAX=NO
```

The messages in the above example indicate that you have the SERNET Keyword Option to start ZDD (ZDD=*portid*), but you do not have a ZDD license.

To correct this problem:

- Remove the ZDD parameter from your SERNET parms and bounce the started task.

    or

- Obtain a ZDD license and rerun LICUPDAT.

Chapter 4
License Test Utility

LICTEST Utility

LICTEST is a utility that simulates a specified Serena application and performs license validation. The LICTEST program is provided in:

> *somnode*.SERCOMC.V*n*R*n*M*n*.LOAD

Program parameters are used to specify which product and version to test, and which language to use for displaying messages.

In addition to performing license checks, LICTEST displays system information about the system on which it is running. This system information includes the following:

- Machine type and model number

- Machine sequence/serial number

- LPAR name and number

- Group name, if LPAR is a member of a group

- System (machine) capacity

- LPAR capacity

- Group capacity, if LPAR is a member of a group

LICTEST displays some diagnostic messages in addition to those that would normally be issued when running the application. The diagnostic messages identify which specific license elements in the license were used.

Running LICTEST

The JCL for running LICTEST is provided in:

> *somnode*.SERCOMC.V*n*R*n*M*n*.CNTL(LICTEST)

Specifying the Parameters

Parameters are specified using the PARM parameter on the JCL EXEC statement. The following example shows the parameters:

```
PARM='PRODUCT=207,VERSION=7.1,LANGUAGE=ENU'
```

You can run LICTEST with no parameters if you only want to display system information about the system on which it is running. In that case, no license checking will be performed.

Specify the parameters as follows:

Parameter	Description
PRODUCT	Specify the numeric product ID. Refer to the names file for valid values. If no product ID is specified, LICTEST will only display system information, such as capacity, LPAR name, and LPAR group, but will not perform any license check.
VERSION	Specify the product version and release number. For example, 7.1.
LANGUAGE	Specify the three-character language code. The default is **LANGUAGE=ENU** for US English. **NOTE** ENU is the only available option at this time. There must be a message module for the specified language. The message modules can be found in *somnode*.SERCOMC.V*n*R*n*M*n*.LOAD. The member names are LICM0xxx, where xxx is the language code.

Specifying the Data Sets

Specify the data set names used in the following DD statements:

DD Name	Description
STEPLIB	Specify the following libraries: ■ The SERCOMC load library, which contains the LICTEST and LICDSN modules. To validate licenses from CSA, use the default LICDSN module (contains a null license DSN) that was delivered in this library. If there is a valid DSN in the LICDSN module, that license file will take precedence over the licenses in CSA. ■ The Language Environment runtime libraries, CEE.SCEERUN and CEE.SCEERUN2, must be specified if they are not included in the system link list.
SERLIC	License file. Specify without any member name. **The SERLIC DD is optional.** To dynamically allocate the license file using the LICDSN module, or to access licenses in CSA, remove this DD statement.
SYSPRINT	Program messages.
SYSNAME	Specify the name of your names file.

Return Codes

Return codes for LICTEST are as follows:

Return Code	Explanation
0	Completed successfully.
4	Completed with warning messages.
8	The product is not licensed, but processing continued.

Return Code	Explanation
12	The product is not licensed, and processing terminated.
16	There was an invalid parameter or other error. Processing terminated.

Appendix A
License Data

Elements

The following table lists the elements of a license document. There is a single Company element and one or more Product elements. The Product element contains one or more Option elements, and may contain multiple Computer elements.

Element	Description
Company	Required. Your company name.
Product	Required. Contains the Serena product number, and optionally the name.
Option	Required. Product options. Contains the following sub-elements: ■ ExpirationDate ■ ExpirationLimit Every product has a base product (Option "0").
ExpirationDate	Expiration date of the option. If not present, the license is permanent.
ExpirationLimit	Maximum expiration date. Required if ExpirationDate is specified in relative format.
Version	Version and release of the product. If not present, the license is valid for all versions.
Computer	Describes the CPU IDs and capacity limits of one or more machines. Contains the following sub-elements: ■ System Capacity ■ Model ■ CpuId If not present, the license is valid for any machine.
MachineType	The machine type on which the product is licensed to run.

Element	Description
MachineModel	The model number on which the product is licensed to run.
MachineSeq	The machine sequence number (serial number) on which the product is licensed to run. This is a hexadecimal number of up to 16 digits. Leading zeros are not required.
Model	The machine type and model number on which the product is licensed to run. The **Model** element is deprecated and is used only for older product versions. Newer product versions use the **MachineType** and **MachineModel** elements instead of **Model**.
CpuId	The CPU ID on which the product is licensed to run. The **CpuId** element is deprecated and is used only for older product versions. Newer product versions use the **MachineSeq** and **MachineType** elements instead of **CpuId**.
SystemCapacity	The maximum machine capacity on which the product is licensed to run. If not present, there is no limit on system capacity.
Lpar	Describes a specific LPAR on which product is licensed to run. Contains the following sub-elements: ■ LPAR Number ■ LPAR Name ■ Group Name ■ LPAR Capacity ■ Group Capacity If not present, the license is valid for any LPAR.

Element	Description
LparNumber	The ID number of an LPAR on which the product is licensed to run. The LPAR number has a value in the range 0 – 255. An LPAR can alternatively be identified with the <LparName> element instead. <LparName> is the preferred method for identifying an LPAR.
LparName	The name of an LPAR on which the product is licensed to run.
GroupName	The name of an LPAR group on which the product is licensed to run. Using this element requires that the LPAR be defined as a member of a group. LPAR groups allow capacity limits to set for a group of LPARs as a whole, rather than setting individual capacity limits for each LPAR.
LparCapacity	The maximum LPAR capacity on which the product is licensed to run. If not present, there is no limit on LPAR capacity.
GroupCapacity	The maximum capacity for the group on which the product is licensed to run. If not present, there is no limit on group capacity. Using this element requires that the LPAR be defined as a member of a group. LPAR groups allow capacity limits to set for a group of LPARs as a whole, rather than setting individual capacity limits for each LPAR.
Users	The maximum number of concurrent users that can use the product. If not present, there is no limit on the number of users.

Element	Description
LicenseType	Required. Values are 0, 1, or 2, and indicate the following types of licenses: ■ 0 Standard ■ 1 Demo ■ 2 Disaster Recovery
WarningDays	Warning days. The number of days before license expiration that warning messages are displayed. If not present, no warning messages will be displayed.
GraceDays	Grace period days. The number of days after license expiration that the product can continue to be used. Second-level warning messages are displayed during the grace period. If not present, no grace period is given.
FailureAction	The type of action to take on failure: ■ 0 Terminate ■ 1 Continue If not present, the default is 0 (terminate).
FailureMsg	The type of message to display on failure: ■ 0 Normal message ■ 1 Highlighted message If not present, the default is 0 (normal).
SeverityFlags	A hexadecimal number representing flags indicating whether specific types of violations are treated as warnings or failures. ■ 1 Treat model number violations as failures If not present, the default is 0 (normal).

Element	Description
LicenseId	License ID. A unique numeric identifier used by Serena.
TimeStamp	Required. Time license issued.
Key	Required. The key used by Serena application programs for validating the license.

Example

In the example that follows, Western Bank is licensed for ChangeMan ZDD and ChangeMan ZMF.

- ChangeMan ZDD is licensed with only the base product (Option ID "0"). This license has an expiration date of 2010/11/21.

- For ChangeMan ZMF, the base product (ChangeMan) and two options (DB2 and ERO) are licensed, all with an expiration date of 2010/11/21.

```xml
<?xml version="1.0"?>
<License>
  <Company>Western Bank</Company>
    <Product ID="210" Name="ChangeMan ZMF">
    <Version>7.2</Version>
    <Option ID="0" Name="ChangeMan ZMF">
      <ExpirationDate>+90</ExpirationDate>
      <ExpirationLimit>2010/12/31</ExpirationLimit>
    </Option>
    <Option ID="4" Name="LoadBalance">
      <ExpirationDate>+90</ExpirationDate>
      <ExpirationLimit>2010/12/31</ExpirationLimit>
    </Option>
    <Computer>
      <MachineType>2094</MachineType>
      <MachineModel>713</MachineModel>
      <MachineSeq>6ABF3</MachineSeq>
      <SystemCapacity>795 MSU</SystemCapacity>
    </Computer>
    <Computer>
      <MachineType>2094</MachineType>
      <MachineModel>403</MachineModel>
      <MachineSeq>54C17</MachineSeq>
      <SystemCapacity>80 MSU</SystemCapacity>
      <Lpar>
        <LparName>DEV001</LparName>
        <LparCapacity>40 MSU</LparCapacity>
      </Lpar>
      <Lpar>
        <LparNumber>7</LparNumber>
        <LparCapacity>50 MSU</LparCapacity>
      </Lpar>
    </Computer>
    <Users>50</Users>
    <Capacity>5000</Capacity>
    <LicenseType>1</LicenseType>
    <WarningDays>30</WarningDays>
    <GraceDays>60</GraceDays>
    <FailureAction>0</FailureAction>
    <FailureMsg>2</FailureMsg>
    <LicenseId>13671</LicenseId>
    <TimeStamp>2009/12/31 23:59:59.299</TimeStamp>
    <Key>7JP2-X4RP-QW9X</Key>
  </Product>
</License>
```

Index

O

overview
   licensing 8

P

parameters
   LOAD 11, 25
problems 26

R

recommendations
   CSA or license file 13
report
   license update 22
return codes 25, 32
runtime libraries
   availability 17
   Language Environment 16

T

testing licenses 30
   return codes 32
troubleshooting 26

U

unloading SER10TY data sets 20
updating
   license file 21
using different versions 8

V

versions, different 8