

---

# Using Platform LSF on Windows

Platform LSF  
Version 8.0  
January 2011



## Copyright

© 1994-2011 Platform Computing Corporation.

Although the information in this document has been carefully reviewed, Platform Computing Corporation ("Platform") does not warrant it to be free of errors or omissions. Platform reserves the right to make corrections, updates, revisions or changes to the information in this document.

UNLESS OTHERWISE EXPRESSLY STATED BY PLATFORM, THE PROGRAM DESCRIBED IN THIS DOCUMENT IS PROVIDED "AS IS" AND WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. IN NO EVENT WILL PLATFORM COMPUTING BE LIABLE TO ANYONE FOR SPECIAL, COLLATERAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES, INCLUDING WITHOUT LIMITATION ANY LOST PROFITS, DATA, OR SAVINGS, ARISING OUT OF THE USE OF OR INABILITY TO USE THIS PROGRAM.

## We'd like to hear from you

You can help us make this document better by telling us what you think of the content, organization, and usefulness of the information. If you find an error, or just want to make a suggestion for improving this document, please address your comments to [doc@platform.com](mailto:doc@platform.com).

Your comments should pertain only to Platform documentation. For product support, contact [support@platform.com](mailto:support@platform.com).

## Document redistribution and translation

This document is protected by copyright and you may not redistribute or translate it into another language, in part or in whole.

## Internal redistribution

You may only redistribute this document internally within your organization (for example, on an intranet) provided that you continue to check the Platform Web site for updates and update your version of the documentation. You may not make it available to your organization over the Internet.

## Trademarks

LSF is a registered trademark of Platform Computing Corporation in the United States and in other jurisdictions.

ACCELERATING INTELLIGENCE, PLATFORM COMPUTING, PLATFORM SYMPHONY, PLATFORM JOB SCHEDULER, PLATFORM ISF, PLATFORM ENTERPRISE GRID ORCHESTRATOR, PLATFORM EGO, and the PLATFORM and PLATFORM LSF logos are trademarks of Platform Computing Corporation in the United States and in other jurisdictions.

UNIX is a registered trademark of The Open Group in the United States and in other jurisdictions.

Linux is the registered trademark of Linus Torvalds in the U.S. and other countries.

Microsoft is either a registered trademark or a trademark of Microsoft Corporation in the United States and/or other countries.

Windows is a registered trademark of Microsoft Corporation in the United States and other countries.

Intel, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

Other products or services mentioned in this document are identified by the trademarks or service marks of their respective owners.

## Third-party license agreements

<http://www.platform.com/Company/third.part.license.htm>

---

# Contents

1	Test Your Platform LSF Installation .....	5
	Check the licenses .....	6
	Check the cluster .....	7
	Check the Load Information Manager (LIM) .....	8
	Check the Remote Execution Server (RES) .....	9
	Platform LSF on Platform EGO .....	10
	Check Platform LSF .....	13
2	Permanent Platform LSF Licenses and FlexNet on Windows hosts .....	15
	Types of Platform LSF licenses .....	16
	FlexNet license management .....	17
	Platform LSF license file (license.dat) .....	18
	Obtain an Platform LSF license .....	19
	Update licenses .....	20
	Create the FlexNet service .....	22
	Start the license server on Windows .....	23
	Check the license server status .....	24
	Install a new permanent license .....	25
3	Platform LSF Default User Mapping .....	27
	About Platform LSF default user mapping .....	28
	Specify user names .....	30
	Configure Platform LSF default user mapping .....	32
	Syntax substitution for Windows user names .....	33
4	Environment .....	35
	Job execution environment .....	36
	Control the execution environment using job starters .....	37
5	Charting Resources with Windows Performance Monitor .....	39
	Platform LSF Monitor statistics .....	40
	Install Platform LSF Monitor .....	42
	Configure Platform LSF Monitor .....	43
	Administer Platform LSF Monitor .....	44
6	Dynamic IP Addressing for Platform LSF Hosts .....	45
	How Platform LSF works with dynamic IP addressing .....	46
	Set up DHCP clients .....	47

7	Displaying a Graphical User Interface in Platform LSF with Microsoft Terminal Services	49
	How Platform LSF works with Terminal Services	50
	Requirements	52
	Register the Microsoft Terminal Services Advanced Client ActiveX Control	53
	Configure Terminal Services for Platform LSF	54
	Configure Platform LSF to run Terminal Services jobs	56
	Submit Platform LSF jobs to Terminal Services hosts (tssub)	57
	Limit the number of Terminal Services jobs on a host	58
	Submit a Terminal Services job from UNIX	59
8	Installing Platform LSF in a Mixed Cluster	61
	Set up a Linux cluster with Windows compute nodes	62
	Set up a Windows cluster with Linux compute nodes	66
	Set up a Windows cluster with Linux compute nodes and EGO controlling Platform LSF daemons	67

## Test Your Platform LSF Installation

Before you make LSF available to users, make sure LSF is installed and operating correctly. You should:

- Check the cluster configuration
- Start the LSF daemons (LSF services)
- Verify that your new cluster is operating correctly

If you have a mixed UNIX and Windows cluster, make sure you can perform operations from both UNIX and Windows hosts.

## Check the licenses

You must have a permanent license. If you have a DEMO license, proceed to checking the cluster status. The FlexNet License Server service is installed as a Windows service to start automatically.

1. Select Start > Settings > Control Panel > Services and make sure the FlexNet License Server service is started.
2. From the command line, check the license server status and display the number of licenses available:

```
C:\lsf\8.0\etc> lmutil lmstat -a -c %LSF_ENVDIR%/license.dat
```

3. From the command line, display the products that are licensed for any host in the cluster:

```
C:\lsf\8.0\bin> lshosts -l hostA
```

# Check the cluster

Before using any LSF commands, wait a few minutes for LSF services to start.

1. Log on to any host in the cluster.
2. Check the configuration files.

C: \LSF\_8. 0> **lsadmin ckconfig -v**

Typical output is as follows:

```
C: \LSF_8. 0>lsadmin ckconfig -v
Checking configuration files ...
Platform EGO 1.2.4, Nov 2 2011
Copyright (C) 1992-2011 Platform Computing Corporation
binary type: nt-x86
Reading configuration from C:\LSF_8.0\conf\ego\cluster1\kernel/ego.conf
Dec 21 08:38:59 2011 4196:1492 6 8.0 Lim starting...
Dec 21 08:38:59 2011 4196:1492 6 8.0 LIM is running in advanced workload execution
mode.
Dec 21 08:38:59 2011 4196:1492 6 8.0 Master LIM is not running in
EGO_DISABLE_UNRESOLVABLE_HOST mode.
Dec 21 08:38:59 2011 4196:1492 5 8.0 C:\LSF_8.0\8.0\etc\lim.exe -C
Dec 21 08:38:59 2011 4196:1492 7 8.0 setMyClusterName: searching cluster files...
Dec 21 08:38:59 2011 4196:1492 7 8.0 setMyClusterName: local host hostA belongs to
cluster cluster1
Dec 21 08:38:59 2011 4196:1492 3 8.0 domanager(): C:\LSF_8.0\conf/
lsf.cluster.cluster1(13):
The cluster manager is the invoker <LSF\lsfadmin> in debug mode
Dec 21 08:38:59 2011 4196:1492 6 8.0 reCheckClass: numhosts 1 so reset exchIntvl to
15.00
Dec 21 08:38:59 2011 4196:1492 7 8.0 getDesktopWindow: no Desktop time window
configured
Dec 21 08:38:59 2011 4196:1492 6 8.0 Checking Done.
-----
No errors found.
```

3. Start the LSF cluster.

- a) If you have a Windows-only cluster, start the LSF cluster:

C: \lsf\8.0\bin> **lsfstartup**

This command starts the LSF services, Platform LIM, Platform RES, and Platform SBD on all LSF Windows hosts. It could take up to 20 seconds.

- b) If you have a mixed UNIX-Windows cluster, you need to log on to a UNIX host and start the UNIX daemons with `lsfstartup`, and then log on to a Windows host and use `lsfstartup` from a Windows host to start LSF services on all Windows hosts.

4. Display the cluster name and master host name:

**lsid**

## Check the Load Information Manager (LIM)

1. Display cluster configuration information about resources, host types, and host models:

### **lsinfo**

The information displayed by `lsinfo` is configured in `LSF_CONFDIR\lsf.shared`.

2. Display configuration information and status of LSF hosts:

### **lshosts**

The output contains one line for each host in the cluster. Type, model, and resource information is configured in the `LSF_CONFDIR\lsf.cluster.cluster_name` file. The `cpuf` matches the CPU factor given for the host model in `LSF_CONFDIR\lsf.shared`.

3. Display the current load levels of the cluster:

### **lsload**

The output contains one line for each host in the cluster. The status should be ok for all hosts in your cluster.



## Check the Remote Execution Server (RES)

You must use your user password using `lspasswd`.

1. Run a command on one LSF host, using the RES:

**`lsrun -v -m hostA hostname`**

2. Run a command on a group of hosts, using the RES:

**`lsgrun -v -m "hostA hostB hostC" hostname`**

3. Check for OK status on cross-cluster configuration information:

**`lsclusters -l`**

## Platform LSF on Platform EGO

LSF on Platform EGO allows EGO to serve as the central resource broker, enabling enterprise applications to benefit from sharing of resources across the enterprise grid.

### How to handle parameters in `lsf.conf` with corresponding parameters in `ego.conf`

When EGO is enabled, existing LSF parameters (parameter names beginning with `LSB_` or `LSF_`) that are set only in `lsf.conf` operate as usual because LSF daemons and commands read both `lsf.conf` and `ego.conf`.

Some existing LSF parameters have corresponding EGO parameter names in `ego.conf` (`LSF_CONFDIR\lsf.conf` is a separate file from `LSF_CONFDIR\ego\cluster_name\kernel\ego.conf`). You can keep your existing LSF parameters in `lsf.conf`, or you can set the corresponding EGO parameters in `ego.conf` that have not already been set in `lsf.conf`.

You cannot set LSF parameters in `ego.conf`, but you can set the following EGO parameters related to LIM, PIM, and ELIM in either `lsf.conf` or `ego.conf`:

- `EGO_DAEMONS_CPUS`
- `EGO_DEFINE_NCPUS`
- `EGO_SLAVE_CTRL_REMOTE_HOST`
- `EGO_WORKDIR`
- `EGO_PIM_SWAP_REPORT`

You cannot set any other EGO parameters (parameter names beginning with `EGO_`) in `lsf.conf`. If EGO is not enabled, you can only set these parameters in `lsf.conf`.

---

#### Note:

If you specify a parameter in `lsf.conf` and you also specify the corresponding parameter in `ego.conf`, the parameter value in `ego.conf` takes precedence over the conflicting parameter in `lsf.conf`.

If the parameter is not set in either `lsf.conf` or `ego.conf`, the default takes effect depends on whether EGO is enabled. If EGO is not enabled, then the LSF default takes effect. If EGO is enabled, the EGO default takes effect. In most cases, the default is the same.

Some parameters in `lsf.conf` do not have exactly the same behavior, valid values, syntax, or default value as the corresponding parameter in `ego.conf`, so in general, you should not set them in both files. If you need LSF parameters for backwards compatibility, you should set them only in `lsf.conf`.

---

If you have LSF 6.2 hosts in your cluster, they can only read `lsf.conf`, so you must set LSF parameters only in `lsf.conf`.

### Platform LSF and Platform EGO corresponding parameters

The following table summarizes existing LSF parameters that have corresponding EGO parameter names. You must continue to set other LSF parameters in `lsf.conf`.

lsf.conf parameter	ego.conf parameter
LSF_API_CONNTIMEOUT	EGO_LIM_CONNTIMEOUT
LSF_API_RECVTIMEOUT	EGO_LIM_RECVTIMEOUT
LSF_CLUSTER_ID (Windows)	EGO_CLUSTER_ID (Windows)
LSF_CONF_RETRY_INT	EGO_CONF_RETRY_INT
LSF_CONF_RETRY_MAX	EGO_CONF_RETRY_MAX
LSF_DEBUG_LIM	EGO_DEBUG_LIM
LSF_DHPC_ENV	EGO_DHPC_ENV
LSF_DYNAMIC_HOST_TIMEOUT	EGO_DYNAMIC_HOST_TIMEOUT
LSF_DYNAMIC_HOST_WAIT_TIME	EGO_DYNAMIC_HOST_WAIT_TIME
LSF_ENABLE_DUALCORE	EGO_ENABLE_DUALCORE
LSF_GET_CONF	EGO_GET_CONF
LSF_GETCONF_MAX	EGO_GETCONF_MAX
LSF_LIM_DEBUG	EGO_LIM_DEBUG
LSF_LIM_PORT	EGO_LIM_PORT
LSF_LOCAL_RESOURCES	EGO_LOCAL_RESOURCES
LSF_LOG_MASK	EGO_LOG_MASK
LSF_MASTER_LIST	EGO_MASTER_LIST
LSF_PIM_INFODIR	EGO_PIM_INFODIR
LSF_PIM_SLEEPTIME	EGO_PIM_SLEEPTIME

## Parameters that have changed in Platform LSF

The default for `LSF_LIM_PORT` has changed to accommodate EGO default port configuration. On EGO, default ports start with `lim` at 7869, and are numbered consecutively for `pem`, `vemkd`, and `egosc`.

This is different from previous LSF releases where the default `LSF_LIM_PORT` was 6879. `res`, `sbatchd`, and `mbatchd` continue to use the default pre-version 7 ports 6878, 6881, and 6882.

Upgrade installation preserves existing port settings for `lim`, `res`, `sbatchd`, and `mbatchd`. EGO `pem`, `vemkd`, and `egosc` use default EGO ports starting at 7870, if they do not conflict with existing `lim`, `res`, `sbatchd`, and `mbatchd` ports.

## EGO connection ports and base port

On every host, a set of connection ports must be free for use by LSF and EGO components.

LSF and EGO require exclusive use of certain ports for communication. EGO uses the same four consecutive ports on every host in the cluster. The first of these is called the base port.

The default EGO base connection port is 7869. By default, EGO uses four consecutive ports starting from the base port. By default, EGO uses ports 7869-7872.

The ports can be customized by customizing the base port. For example, if the base port is 6880, EGO uses ports 6880-6883.

LSF and EGO needs the same ports on every host, so you must specify the same base port on every host.

# Check Platform LSF

The LIM and mbatchd must be running on the master host and on the submission host (the host from which you run the command).

1. Verify the LSF daemon configuration:

```
C:\LSF_8.0>badmin ckconfig -v
```

The message No errors found. displays.

2. Run some basic commands and check the status: OK (hosts) and Open:Active (queues):

```
bhosts
```

```
bqueues
```

3. Display the default queue:

```
C: \l sf \b i n>bparams
```

4. Submit a test job to the default queue named normal :

```
C: \l sf \8. 0 \b i n> bsub sleep 60
```

Job <1> is submitted to default queue <normal>.

5. Display the job status:

```
C: \l sf \8. 0 \b i n> bjobs
```

If all hosts are busy, the job is not started immediately and the STAT column says PEND. The job sl eep 60 should take one minute to run. When the job completes, LSF sends mail reporting the job completion.



## Permanent Platform LSF Licenses and FlexNet on Windows hosts

Managing LSF licenses is described in *Administering Platform LSF*. This chapter includes additional information that is necessary to license a cluster that includes Windows hosts. You must have a valid license to run LSF.

If you install a permanent license using the LSF installer, and you do not already use FlexNet software, you must install and configure FlexNet before running the LSF installer.

## Types of Platform LSF licenses

You can run LSF with either an DEMO (evaluation) or a permanent license.

DEMO (evaluation) licenses	If you have not yet purchased LSF, get a demo license key. Typically, a demo license licenses all LSF products and expires after 30 days. If you purchase LSF during this time, you can switch to a permanent license with no interruption in service.
Permanent licenses	A permanent license restricts the total number of hosts in the cluster and defines which LSF products will be used. It defines which hosts in your network will run LSF, and how they are arranged into clusters. This type of license does not expire, but has to be updated if you add more hosts or add another LSF product.



# FlexNet license management

Platform LSF uses the FlexNet license management software from Acresso Software.

LSF is compatible with FlexNet version 10.8.5.0 or later. If you run an older version, you must upgrade FlexNet before you can use LSF.

## FlexNet license server

A FlexNet license server is required for permanent licenses only. Permanent LSF licenses are managed by the FlexNet license server daemon (`lmgrd`) running on one or three hosts in your network.

If you install a permanent license using the LSF installer, and you do not already use FlexNet software, you must install and configure FlexNet before running the LSF installer.

To get a permanent license, you must provide information about the license server host. Platform Computing requires the hardware host name and host identifier for each license server host at your site.

Choose a reliable host as the FlexNet license server to ensure that the LSF licenses are always available. In many installations, the license server runs on the host that is the dedicated file server for the Platform LSF software. This permits the licenses to be available whenever LSF is available.

You can specify 3 hosts to be license servers instead of just one.

## Existing FlexNet server

If you already have a FlexNet license server installed, you can merge the contents of your LSF license file into your existing FlexNet license file.

## FlexNet license management tools

On Windows, FlexNet commands are subcommands of `lmutil`.

For example, to retrieve the FlexNet host ID, run `lmutil lmhostid` (on UNIX, run `lmhostid`). To check the license server, run `lmutil lmstat` (on UNIX, run `lmstat`).

## Platform LSF license file (license.dat)

The LSF license is stored in a text file. By default, the file name is `license.dat`. Your permanent license file must contain all the products that are defined in `LSF_CONFDIR\lsf.cluster.cluster_name`.

For a description of the license file format, see *Administering Platform LSF*.

## Locations of the license file

By default, there are two copies of your permanent license file. The installer copies your LSF license to the LSF directory, but you must update the FlexNet directory manually. After the permanent LSF license is installed, you have to update both copies when you make changes to your LSF licensing.

- LSF uses the file specified by `LSF_LICENSE_FILE` in `lsf.conf`. By default, the file is `license.dat` in `LSF_CONFDIR`. For example, if your LSF installation directory is `lsfshare\lsf` on HostA, your `lsf.conf` file might include the following line:  

```
LSF_LICENSE_FILE=\\HostA\lsfshare\lsf\conf\license.dat
```
- The FlexNet license manager uses the file in the FlexNet installation directory on the host that is the FlexNet license server. By default, the file is `license.dat` in `C:\flexlm`. For example, your license server might include the following file:

```
C:\flexlm\license.dat
```

## Using FlexNet to manage other software

If you use FlexNet to manage other software, the `license.dat` file used by FlexNet contains information that is not related to LSF, but the `license.dat` file used by LSF only contains LSF information. Even if the files in both locations are not identical, the lines relating to LSF must be exactly the same.

## Modifying the Platform LSF license

- On Windows, you must always modify the file used by FlexNet manually. The installer does not overwrite the file `C:\flexlm\license.dat` if it already exists.
- If you move the license file used by LSF, you must update the `LSF_LICENSE_FILE` parameter in `lsf.conf`.

# Obtain an Platform LSF license

Where to get it:

- Contact your LSF vendor.
- Complete the evaluation form on the Platform Web site ([www.platform.com](http://www.platform.com)).

To allow Platform or your LSF vendor to create or update a permanent license, you must provide the following information:

- Host name of the license server
- FlexNet host ID of the license server
- Number of LSF license units

## Get the host name

To retrieve the host name of the license server:

1. Log on to the host that is going to be your license server.
2. Open a command prompt (Start | Programs | Command Prompt).
3. Type the host name command:

```
hostname
```

Windows displays the hardware host name.

## Get the FlexNet host ID

To retrieve the FlexNet host ID of the license server, run the `lmutil lmhostid` command at an MS-DOS prompt:

```
C:\temp>%LSF_SERVERDIR%\lmutil lmhostid
lmutil - Copyright (C) 1989-2010 Flexera Software, Inc.
The FlexNet host ID of this machine is "0090273eeb99"
```

# Update licenses

Once you have created an LSF cluster, you can update the licensing at any time. The procedure varies:

## Update an evaluation cluster

If you have a demo license, you must update it to a permanent license when you finish the evaluation and purchase LSF. LSF permanent licenses use FlexNet license management software.

1. If you already use FlexNet, check the version. LSF is compatible with FlexNet version or later. If you have an older version, you must upgrade FlexNet before installing the permanent LSF license.
2. If you do not have FlexNet installed, learn about FlexNet and permanent licensing.
3. Obtain a new permanent license.
4. If you already have FlexNet installed, you must update your FlexNet license file manually. Append the entire contents of the new LSF license to your existing file.
5. If you have not yet installed FlexNet, you must install FlexNet, then set up the permanent license file used by FlexNet.

## Expand an existing cluster

When you have a permanent LSF license, the number of hosts and selection of LSF products and features is controlled by licensing.

- To expand the size of the cluster by adding hosts, you probably need to purchase additional licenses for the additional hosts.
- To enable new LSF products or features such as LSF MultiCluster, you need to purchase additional licenses. Sometimes you also need to install additional software. Update your license before you update the PRODUCTS line in the Parameters section of `LSF_CONFDIR\lsf.cluster.cluster_name`.

To make a change that involves additional licensing, the first step is to manually update your existing license files. In the license file used by FlexNet, the lines related to LSF must be identical to the contents of the license file used by LSF.

1. Obtain a new permanent license.

Instead of getting an entire new license, you will probably get an LSF INCREMENT license key or a new FEATURE line. Append the new lines to the existing LSF content without overwriting or deleting any information. If you have INCREMENT lines, you must add them immediately after the FEATURE line for the product.

2. Update FlexNet.
3. Update LSF.

## Upgrade to a new version of Platform LSF

Sometimes a minor upgrade or patch does not require any changes to the licensing. If the upgrade requires a new license, update the FlexNet license manually and follow the steps in *Migrating Platform LSF Version 7 to Platform LSF Version 8 on Windows*.

## Update Platform LSF license features

1. Open the license file using a text editor such as Notepad.

The license file used by LSF is specified by `LSF_LICENSE_FILE` in `lsf.conf`.

2. Make the required changes. This normally involves appending the new lines without overwriting or deleting existing content.
3. Save and close the file.
4. Reconfigure LSF:

```
lsadmin reconfig
```

## Update a FlexNet license

1. Open the license file using a text editor.

The license file used by FlexNet is normally `C:\flexlm\license.dat` on the license server host.

2. Make the required changes.

This normally involves adding the new lines to your FlexNet license. The only time you replace existing lines is when you upgrade from one version of LSF to another. Do not overwrite or delete any content that is not related to LSF.

3. Save and close the file.
4. Update FlexNet. Log on to the license server host and re-read the updated license file:

```
C:\flexlm\bin> lmutil lmread -c C:\flexlm\license.dat
```

You see a message similar to the following:

```
lmread - Copyright (C) 1989-2010 Flexera Software, Inc.
lmread successful
```

## Create the FlexNet service

If you are using a permanent LSF license, you must run `install s.exe` to install or remove the FlexNet license manager as a Windows Service. `install s.exe` is located under `LSF_TOP\8.0\etc`.

The `install s` command has the following syntax:

```
installs -c license_file_path -e path_to_lmgrd.exe -l log_file_path -n service_name [-k lmgrd_parameters]
```

If `-n` is not specified, `FlexNet License Manager` is used as the service name.

The `-k` switch is optional and is used to pass one or more startup command-line parameters (`-local`, `-x lmdown`, and `-x lmremove`) to `lmgrd`.

You must use `-l log_file_path` to specify the path to the `lmgrd.log` file.

See the *FlexNet Reference Manual* for more information about the `install s` command.

## Install the FlexNet service

Run `install s.exe` to install or remove the FlexNet license manager as a Windows Service.

1. Open a Windows Command Prompt.
2. Run the `install s` command.

```
LSF_TOP\8.0\etc\install s -c license_file_path -e path_to_lmgrd.exe -l log_file_path
```

For example:

```
C:\LSF_8\8.0\etc\install s -c C:\sharetest\license.dat -e C:\LSF_8\8.0\etc\lmgrd.exe  
-l C:\sharetest\logs\lmgrd.log
```

The `lmgrd.exe` is located under `LSF_TOP\8.0\etc`.

## Remove the FlexNet service

If you need to remove the FlexNet license manager service, run `install s.exe`.

1. Open a Windows Command Prompt.
2. Run the `install s` command.

```
LSF_TOP\8.0\etc\install s.exe -r -n "service_name"
```

For example:

```
LSF_TOP\8.0\etc\install s.exe -r -n "FlexNet License Manager"
```

If you did not specify `-n` when you installed the license manager service, `FlexNet License Manager` is used as the default service name.

## Start the license server on Windows

### Start FlexNet automatically

1. Restart the license server host.

### Start FlexNet manually

1. Log on to the license server host.
2. Choose Services in the Windows Control Panel.
3. Choose Services in the Windows Control Panel.

## Check the license server status

If you are using a permanent LSF license, check the license server by starting the license manager and displaying the status with the `lmutil lmstat` command on Windows.

If you are using a demo license, you do not have to perform this check because a demo license does not run a license server daemon.

1. Start the license server.
2. Check the status of the license server and display the number of licenses available. For example:

```
C:\flexlm\bin> lmutil lmstat -a -c C:\flexlm\license.dat
```

The output of `lmstat` gives the status of the license server together with the name of the LSF vendor daemon (`lsf_d`). The usage information for each product in the license file is provided



# Install a new permanent license

If you do not already use FlexNet, you must make sure FlexNet is installed and running properly.

Complete the following steps to replace a temporary demo license used by LSF with a permanent license.

1. Shut down the master host.

```
lsfshutdown -m master_host_name
```

2. Copy the path to the permanent license to LSF\_LICENSE\_FILE defined in `lsf.conf`.

If you installed LSF with a default installation, the license file is installed in the LSF configuration directory (`LSF_CONFDIR\license.dat`).

---

**Note:**

You can put the permanent license anywhere you want, as long as you update the LSF\_LICENSE\_FILE parameter in `$LSF_CONFDIR\lsf.conf`.

---

3. Start the master host.

```
lsfstartup -m master_host_name
```



## Platform LSF Default User Mapping

The default user mapping in LSF has no effect on a UNIX-only cluster. You do not need to understand this feature unless your cluster includes Windows hosts.

## About Platform LSF default user mapping

The default user mapping determines whether you can specify a Windows user in LSF by the user name alone. In a mixed cluster, it also specifies whether a Windows user account maps to a UNIX account of the same name, to allow cross-platform operation.

## How Platform LSF default user mapping works

If you specify an LSF user domain, the default user mapping is enabled. For a multiple-domain Windows environment on a UNIX-Windows mixed cluster, you can specify an unlimited number of Windows domains as the LSF user domain.

When the default user mapping is enabled,

- A user name specified without a domain is interpreted (on a Windows host) as belonging to the LSF user domain
- A user name specified with the domain name of the LSF user domain is stripped of the domain name

## Mixed cluster

In a mixed UNIX-Windows environment, if your Windows account in the LSF user domain has the same user name as your UNIX account, LSF's default user mapping lets LSF schedule and track jobs from both accounts as if they belong to a single user. On the execution host, LSF automatically runs the job using whichever of the two accounts is appropriate for that host.

To submit cross-platform jobs when your accounts have different user names in different environments, you should configure user account mapping for individual users.

## Multiple domain accounts

To run jobs, the existing domain trust relationships apply in LSF, so if the execution domain trusts the submission domain, your job can run in the execution domain under your submission account.

If a user domain is...	Then LSF treats the Windows and UNIX user as...
Specified by the parameter LSF_USER_DOMAIN	The same user
Not specified by the parameter LSF_USER_DOMAIN	Different users

Accounts with the same user name in different domains are still treated as separate users by LSF.

You can use the environment variable LSF\_EXECUTE\_DOMAIN to specify only one of the domains listed in LSF\_USER\_DOMAIN. When you specify an execution domain, LSF runs the job using the specified domain user account, without trying all of the domain accounts in the order listed in LSF\_USER\_DOMAIN.

## Local accounts

If your local account has the same user name and password on every Windows host, LSF's default user mapping lets LSF schedule and track jobs from all hosts as if they belong to a single user. On the execution host, LSF automatically runs the job using the local user account.

If your accounts have different user names in different environments, you should configure user account mapping.

## Installation examples

In the following examples, assume you are User1, and you have a valid user account in 3 Windows domains as well as a valid UNIX account. Not all the accounts can be used with LSF. Depending on the type of cluster, and the way you install the cluster, here are the different ways that LSF is configured:

### Install or upgrade a UNIX-only cluster

No mapping. You have one UNIX account, and LSF recognizes 1 user:

- user 1 (UNIX account)

### Install a new Windows-only cluster

No mapping. You have 3 Windows accounts. For purposes of fairshare, per-user job slot limits, displaying statistical data, and so on, LSF recognizes 3 separate users:

- DOMAIN A\user 1
- DOMAIN B\user 1
- DOMAIN C\user 1

### Create a new UNIX-Windows cluster

You must enable default user mapping for one of your Windows accounts (such as Domain A) so that you can run cross-platform jobs between UNIX and Windows. LSF recognizes 3 separate users:

- user 1 (your UNIX and Domain A accounts are treated as a single LSF user)
- DOMAIN B\user 1
- DOMAIN C\user 1

If you never run cross-platform jobs, you might choose to disable default user mapping by not specifying an LSF user domain. LSF then recognizes 4 separate users:

- user 1 (UNIX account)
- DOMAIN A\user 1
- DOMAIN B\user 1
- DOMAIN C\user 1

You can specify multiple domains when you define LSF\_USER\_DOMAIN, which will allow users to submit jobs from a UNIX host in a multiple-domain Windows environment.

## Specify user names

In a Windows cluster or mixed UNIX-Windows cluster, in a domain environment, LSF users in different Windows domains might have the same user name. Because of this, LSF uses the Windows domain name with the user name, to differentiate the users.

### User name only

When the default mapping is enabled, the user name alone specifies a user in the LSF user domain. The combination of a user name plus the domain name of the LSF user domain is not used in LSF.

### Domain name with user name

#### Default mapping disabled

All Windows user accounts are specified using the domain name with the user name. There is no LSF user domain.

#### Default mapping enabled

User accounts in all domains except for the LSF user domain are specified using the domain name with the user name.

## How to specify a user name with a domain name

Unless a Windows user account belongs to the LSF user domain (`LSF_USER_DOMAIN` in `lsf.conf`), the combination of domain name and user name specifies a Windows domain user in LSF. The syntax is:

```
[ DOMAIN_NAME/. ]\user_name
```

Type the domain name in capital letters. Use a period ( `.` ) instead of a domain name to specify a local account instead of a domain account.

UNIX systems interpret the single backslash as a special character, so on UNIX you have to use a double backslash to specify the domain name in the command line:

### Windows

```
bjobs -u MYDOMAIN\user1
```

### UNIX

```
bjobs -u MYDOMAIN\\user1
```

## View user names

Use `bjobs -w` to view information about jobs and see the full name of a Windows user, including domain name.

When you run `bjobs`, the default is to truncate user names, and display the names of Windows users without the domain name.

## Windows user authentication

LSF recognizes UNIX and Windows authentication environments, including different Windows domains and individual Windows workgroup hosts.

- In a Windows domain environment, user accounts are validated at the domain level, and your user account is valid on all hosts in your domain (and might be valid in other domains, if there is a trust relationship).
- In a Windows workgroup environment, each host authenticates the user account, so your local account is only valid on one host.

## lspasswd command

You must use `lspasswd` or `wgpaswd` to register and update user names and passwords. The password must be 23 characters or less.

You can run `lspasswd` on Windows in a non-shared file system environment. You must define the parameter `LSF_MASTER_LIST` in `lsf.conf` so that jobs will run with the correct permissions. If this parameter is not defined, LSF assumes that the cluster uses a shared file system environment.

You can also run `lspasswd` to check that the password is valid for the specified user, or to remove a user entry from the password database.

## Password problem notification on Windows

A Windows job may not be able to run because of a problem with the user's LSF password (entered and updated using `lspasswd`). If LSF does not recognize the password, the problem could be:

- The Windows user account password was never registered with LSF with `lspasswd`.
- The password in Windows changed but was not updated in LSF with `lspasswd`.

If a job is in PEND state and LSF cannot run it because of a password problem, by default, LSF puts the job into USUSP and then notifies the user via email. The user can fix the problem, and then use `brsume` to release the job from USUSP.

## Configure Platform LSF default user mapping

Whenever you make any change to default user mapping, you affect users in the old LSF user domain and in the new LSF user domain. If you specify a new LSF user domain, users in both domains will have to use `lspasswd` to register their new names and passwords.

If users in the old and new LSF user domain have the same user name (such as `olddomain\user1` and `newdomain\user1`), then the `user1` account is already registered with LSF, and the user from the new LSF user domain has to change the password. To change the password, he must input the current password, which was set by the old user.

1. To enable or modify default user mapping after you install LSF, set `LSF_USER_DOMAIN` in `lsf.conf` and specify the LSF user domain:

```
LSF_USER_DOMAIN=DomainA
```

- You can also specify multiple domains:

```
LSF_USER_DOMAIN=DomainA: DomainB: DomainC
```

Depending on the cluster configuration, you might have to redefine the service accounts, cluster administrators, queue administrators, user group memberships, and so on, so that your cluster remains operational after you restart the cluster.



## Syntax substitution for Windows user names

In *Administering Platform LSF* and other LSF documentation, a user name is represented by the syntax:

*user\_name*

If your cluster includes Windows hosts, the full syntax for a user account on Windows is:

*[DOMAIN\_NAME /.]user\_name*

Always type the domain name in capital letters.

## Platform LSF commands

In the following LSF commands, use the full syntax to specify a user name.

- bchkpnt
- bdel
- bhi st
- bj obs
- bki ll
- bmi g
- bmod
- brequeue
- bresume
- bstop
- bsub
- bswi tch
- busers
- lsacct
- lspasswd

## Platform LSF files

In the following configuration files and parameters, use the full syntax to specify a user name.

- lsb. hosts
  - USER\_SHARES
- lsb. params
  - SYSTEM\_MAPPING\_ACCOUNT
- lsb. queues
  - ADMINISTRATORS
  - FAIRSHARE
  - USERS
- lsb. users
  - GROUP\_MEMBER
  - USER\_SHARES
  - USER\_NAME
  - LOCAL

## Platform LSF Default User Mapping

- REMOTE
- `lsf.cluster.cluster_name`
  - ADMINISTRATORS
- `lsf.conf`
  - LSF\_SHELL\_AT\_USERS
- `lsf.sudoers`
  - LSF\_EAUTH\_USER
  - LSF\_EEXEC\_USER
  - LSF\_STARTUP\_USERS
  - LSB\_PRE\_POST\_EXEC\_USER

# 4

## Environment

## Job execution environment

By default, Platform LSF transfers environment variables from the submission to the execution host. However, some environment variables do not make sense when transferred.

## How Platform LSF sets the job execution environment

When submitting a job from a Windows to a UNIX machine, the `-L` option of `bsub` can be used to reinitialize the environment variables. If submitting a job from a UNIX machine to a Windows machine, you can set the environment variables explicitly in your job script.

## PATH environment variable on UNIX and Windows

Platform LSF automatically resets the `PATH` on the execution host if the submission host is of a different type. If the submission host is Windows and the execution host is UNIX, the `PATH` variable is set to `/bin:/usr/bin:/sbin:/usr/sbin` and the path to the LSF `bin` directory is appended to it.

If the submission host is UNIX and the execution host is Windows, the `PATH` variable is set to the system `PATH` variable with the path to the LSF `bin` directory appended to it. Platform LSF looks for the presence of the `WINDIR` variable in the job's environment to determine whether the job was submitted from a Windows or UNIX host. If `WINDIR` is present, it is assumed that the submission host was Windows; otherwise, the submission host is assumed to be a UNIX machine.

## Environment variable handling on Windows

The following Windows environment variables are overridden based on the values on the execution host:

- COMPSPEC
- COMPUTERNAME
- NTRESKIT
- OS2LIBPATH
- PROCESSOR\_ARCHITECTURE
- PROCESSOR\_LEVEL
- SYSTEMDRIVE
- SYSTEMROOT
- WINDIR

If the `WINDIR` on the submission and execution host are different, then the system `PATH` variable on the execution host is used instead of that from the submission host.

Avoid using drive names in environment variables (especially the `PATH` variable) for drives that are connected over the network. It is preferable to use the UNC form of the path. This is because drive maps are shared between all users logged on to a particular machine. For example, if an interactive user has drive `F:` mapped to `\\serverX\share`, then any batch job will also see drive `F:` mapped to `\\serverX\share`. However, drive `F:` might have been mapped to a different share on the submission host of the job.

Job starters can be used to perform more site-specific handling of environment variables.

# Control the execution environment using job starters

If running jobs on a Windows execution host, you cannot use the command `bsub -L`. Instead, LSF provides two job starters that apply the user environment of the execution host.

By default, the job starter executables are installed in `LSF_BINDIR`. If you prefer to store them elsewhere, make sure they are in a directory that is included in the default `PATH` on the execution host.

For example, on Windows, put the job starter under `%WINDIR%`.

The source code for the job starters is installed in `LSF_TOP\8.0\examples`.

- Use either of the two following starter scripts to run jobs on a Windows execution host:
  - `preservestarter`: Preserves the default user environment of the execution host; does not include any submission host settings
  - `augmentstarter`: Augments the default user environment of the execution host by adding settings from the submission host that are not already defined on the execution host



## Charting Resources with Windows Performance Monitor

LSF integrates with Windows Performance Monitor, so you can chart LSF cluster, host, queue, and job performance information. Windows Performance Monitor can also be used to trigger external commands when specified thresholds are exceeded.

A service called LSF Monitor passes information from LSF to the Windows Performance Monitor. LSF Monitor must be installed separately.

## Platform LSF Monitor statistics

Once installed, LSF Monitor automatically sends information to the Windows Performance Monitor. Use the Windows Performance Monitor to chart LSF performance information.

The host, queue, and job objects support multiple instances.

The following LSF information is available:

- Cluster information
- Host information
- Queue information
- Job information
- External information

### Cluster information

- Number of available servers
- Number of unavailable servers
- Number of servers where an LSF daemon (sbatchd or RES service) is down
- Number of unlicensed servers
- Number of pending jobs in the cluster
- Number of running jobs in the cluster
- Number of suspended jobs in the cluster
- Number of sick jobs (jobs submitted with no password, jobs with job dependency never satisfied, and jobs pending more than 3 days)
- Response time of LIM (as measured by the time to make an `ls_load` call)
- Response time of `mbatchd` (as measured by the time to make an `lsb_queueinfo` call)

### Host information

- Load indices: `r15s`, `r15m`, `mem`, `swap`, `pg`, `ut`
- Number of running jobs
- Number of suspended jobs
- Number of reserved job slots
- External load Indices

### Queue information

- Number of pending jobs
- Number of running jobs
- Number of suspended jobs
- Number of reserved job slots

### Job information

- CPU time used by the job
- Memory used by the job (for jobs running on UNIX only)
- Swap space used by the job (for jobs running on UNIX only)



## External information

- Values of one or two external load indices (configured by the LSF administrator)

## Install Platform LSF Monitor

You must have a cluster running LSF version 4.0 or higher. You must install LSF Monitor on any LSF server or client host running Windows. The cluster can include UNIX hosts. You must specify a cluster administrator account and password.

The LSF Monitor setup program is installed with LSF. Use `lsfmon -install` to actually install the LSF Monitor service:

1. Log on to a Windows host as an LSF user in an existing LSF cluster.
2. In a command prompt, type:  
**lsfmon -install**  
LSF Monitor is installed.
3. On the Windows Control Panel, click Services.  
The Services window opens.
4. Right-click LSF Monitor and click Properties.
5. In the Log On As section, deselect System Account, select This Account, and specify an LSF cluster administrator account (such as Administrator).
6. Type in the password twice and click OK.
7. In the Services window, select LSF Monitor and click Start to start the service.

# Configure Platform LSF Monitor

Back up your registry before you make any changes.

You can configure sample intervals for host, queue and job information along with external load indices.

LSF Monitor periodically samples information from LSF and updates the Windows Performance Monitor.

By default, information is sampled at the following intervals:

- Host information = 30 seconds
- Queue information = 45 seconds
- Job information = 60 seconds

1. Change the sample intervals for LSF host, job, or queue information by modifying the Windows Registry settings.

- a) Select the Registry subkey:

`HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\LSFMonitor`

- b) Edit the appropriate value, and specify the new sample interval in seconds:

- SampleIntervalHost
- SampleIntervalJob
- SampleIntervalQueue

2. Configure LSF Monitor to monitor external load indices.

- a) Go to the Registry subkey `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\LSFMonitor`.
- b) Specify the appropriate value and type the name of an external load index that is configured in your cluster:
  - ExternalLoadIndex1
  - ExternalLoadIndex2

## Administer Platform LSF Monitor

- Start or stop LSF Monitor.

Use the Windows Control Panel to start or stop the LSF Monitor service.

- Use the Windows Event Viewer to view the Windows event log.

Errors related to LSF API calls and the operation of LSF services are logged to the Windows event log.

- Uninstall LSF Monitor. From a command prompt, type:

**lsfmon -remove**

This command stops the LSF Monitor service if it is running, then removes it and removes related information from the Windows Registry.

## Dynamic IP Addressing for Platform LSF Hosts

### About dynamic host configuration protocol (DHCP)

DHCP (Dynamic Host Configuration Protocol) enables individual computers on an IP network to extract their configurations from particular machines (DHCP servers) that have no exact information about the individual computers until they request the information. This reduces the work necessary to administer a large IP network. The most significant piece of information distributed in this manner is the IP address.

## How Platform LSF works with dynamic IP addressing

LSF hosts running Windows can be configured as DHCP clients, which means their IP address is dynamic. Users who dial in or connect from a remote location might be assigned a different IP address with each connection.

The DHCP server issues an IP address to the LSF host, but LSF gets the IP address from DNS (Domain Name System). A WINS (Windows Internet Naming Service) server synchronizes information between the DHCP and DNS servers.

The IP address should not be changed while there are active TCP/IP connections with the host, for example, while installing LSF or running LSF commands. Normally, the IP address is maintained until the host is restarted or until the network connection is broken.

If an LSF client host is assigned a new IP address, wait for WINS to update DNS before using that host to run LSF.

## Platform LSF client hosts

LSF client hosts can be DHCP clients and can change their IP addresses anytime in a running cluster.

## Platform LSF server hosts

Installing dynamic hosts on Windows allows support for dynamic IP addressing for LSF server hosts using DHCP.

LSF server hosts can be DHCP clients and can change their IP addresses anytime in a running cluster. The master host also saves the slave host IP address.

# Set up DHCP clients

To use DHCP with LSF, your system must include all of the following:

- DHCP server
- WINS server
- DNS server
- LSF hosts acting as DHCP clients

Install Microsoft DNS server and WINS server on the same machine.

1. Configure a short cache timeout value on the WINS server.

The Cache Timeout Value for the WINS Lookup of the DNS should be as short as possible (the 10-minute default may be acceptable, but this should not be increased).

2. Enable dynamic IP addressing for the LSF cluster.

- a) Configure the following parameter in `lsf.conf`: `LSF_DHCP_ENV=Y`
- b) Reconfigure the cluster:

```
lsadmin reconfig  
badmin reconfig
```

LSF checks for any configuration errors. If no fatal errors are found, you are asked to confirm reconfiguration. If fatal errors are found, reconfiguration is aborted.





## Displaying a Graphical User Interface in Platform LSF with Microsoft Terminal Services

# How Platform LSF works with Terminal Services

It is possible to run jobs to display GUI (graphic user interface) on remote hosts in LSF by using Microsoft Terminal Services.

## Environment variables

The following environment variables are available for Terminal Services jobs:

### LSF\_LOGON\_DESKTOP

When `LSF_LOGON_DESKTOP=1`, jobs run in interactive foreground sessions. This allows GUIs to be displayed on the execution host. If this parameter is not defined, jobs run in the background.

### LSB\_TSJOB

When the `LSB_TSJOB` variable is defined to any value, it indicates to LSF that the job is a Terminal Services job.

### LSF\_TS\_LOGON\_TIME

Specifies the time to create a Windows Terminal Service session. You should configure `LSF_TS_LOGON_TIME` according to your network environment.

The default, 300000 milliseconds is suitable for most environments. For a congested network, set `LSF_TS_LOGON_TIME=1000000`

## Job submission

1. Submit the job with `tssub` instead of `bsub`. `tssub` is a wrapper around the `bsub` command which only submits jobs to hosts that have the `mst.s` resource.
2. `tssub` sets the `LSB_TSJOB` and `LSF_LOGON_DESKTOP` environment variables. These variables are then transferred to the execution host.
  - If the job is dispatched to a host in which Terminal Services is not installed or properly configured, the job is set to the `PEND` state and a pending reason is written in `sbatchd.log`. *host\_name*.
  - If `tssub -I` is specified, a terminal display is visible on the submission host after the job has been started.
  - If the job is not a GUI job, LSF runs a command window and output is displayed in the command window when something is written to `stdout`.
  - Pre- and post-execution commands are executed within the terminal session. The job does not complete until post-execution commands complete.
3. View job output with the command `tspeek`. If the terminal window is closed, the job remains running. You can reconnect to view the job with `tspeek`.

## Limitations

- A job submitted as a Terminal Services job cannot be modified to become a non-Terminal Services job with `bmod`.
- The `bsub` option `-o out_file` is not supported for `tssub`.
- Compound resource requirements are not supported for Terminal Services jobs.

- Only Windows bsub options are supported for `tssub`. For example, you cannot use the options `-Ip`, `-IS`, `-L login_shell` of bsub with `tssub`.
- If user mapping is defined, the user who invokes `tspeak` must have the required privileges to access the session.
- MultiCluster is not supported.

## Requirements

- All submission hosts must have the Terminal Services Full Client Windows Installer (MSI) package installed. This package contains the required Microsoft Terminal Services Advanced Client ActiveX Control. Download it from the Microsoft web site.
- All execution server hosts must have the Terminal Services and Terminal Services Full Client Windows Installer (MSI) packages installed.
- Terminal Services server hosts can be in a different domain from submission hosts as long as the user can be authenticated by both domains.
- The service account on Terminal Services server hosts must be a member of the local administrators group.
- Your LSF cluster must be working properly.
- All submission (client) hosts need the ActiveX control, but you don't necessarily have to use the specified MSI, it can also be there if Remote Desktop Connection software is installed (part of the OS installation).
- To run applications from a Terminal Services client, the FEATURE lines in your LSF license must contain the TS\_OK keyword. FlexNet detects when a node-locked uncounted license is running under Windows Terminal Server. Without TS\_OK, a user running on a Terminal Server client is denied a license. See the *FlexNet End User Guide* for more information.

# Register the Microsoft Terminal Services Advanced Client ActiveX Control

You must be a Local Administrator on the host to register the control.

For clients running Windows 2000 or earlier with Remote Desktop Connection, the necessary ActiveX control is installed, but it is not registered by default. The LSF `tssub` command registers the control automatically.

The problem does not occur with the Terminal Service Full Client installation, or on newer versions of Windows or Windows XP.

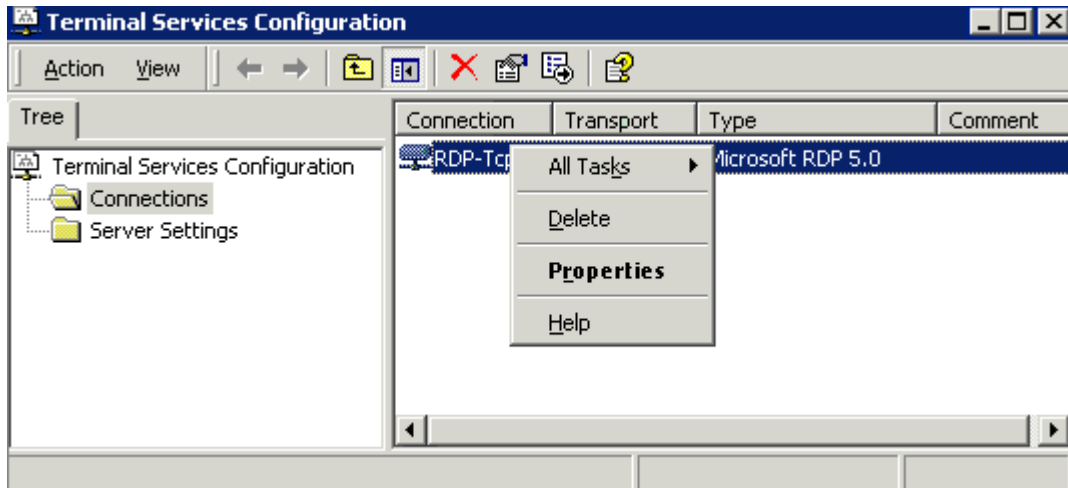
- Submit any job with `tssub`. If you have administrator privileges, `tssub` registers the ActiveX control automatically.
- Register the ActiveX control manually. Type the following Windows command at an MS-DOS prompt:

```
C: \regsvr32.exe mstscax.dll
```

Once the control is registered on the host, the error does not occur. Non-administrator users can run `tssub` with no errors.

# Configure Terminal Services for Platform LSF

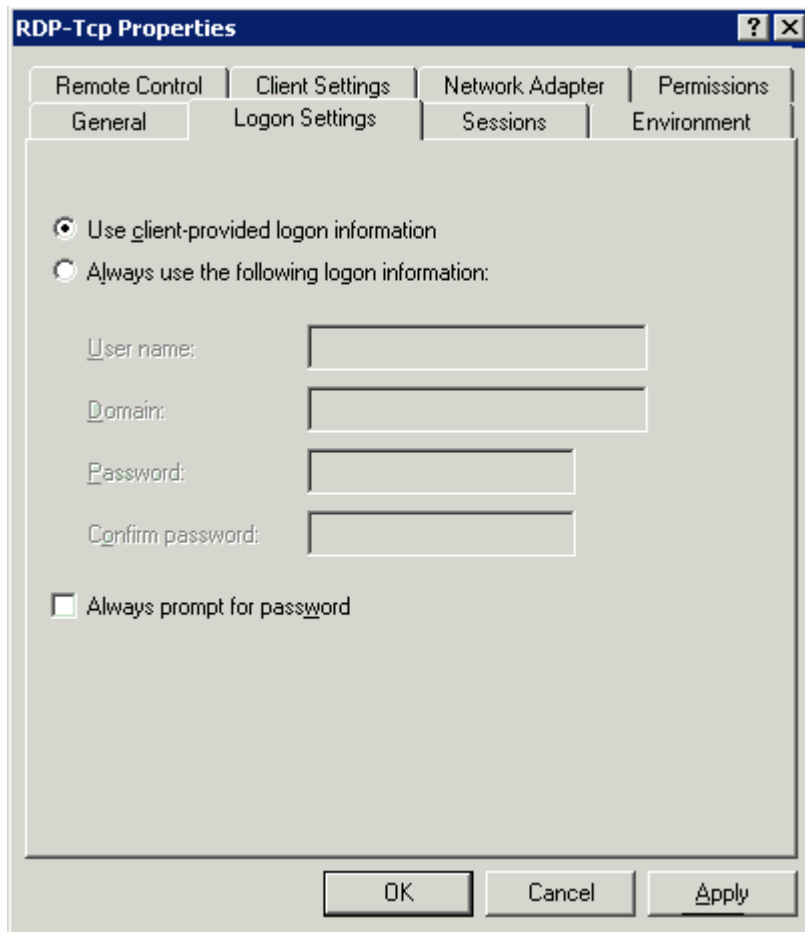
1. Start the Microsoft Management Console (MMC) Terminal Services Configuration snap-in (Start > Programs > Administrative Tools > Terminal Services Configuration).



2. Right-click the configuration for which you want to disable the default password setting, and select Properties.
3. Change the logon information.
  - a) Select Logon Settings > Use client provided logon information.

This ensures you are not using a predefined user to log on to Terminal Services.
  - b) Clear the Always prompt for password check box.

Future connections no longer force a password entry.



4. Click Apply, then click OK and close the dialog.

# Configure Platform LSF to run Terminal Services jobs

1. Define the msts static resource.

Edit `LSF_CONFDIR\lsf.shared` and define the `msts` static resource. Note that the resource name must be `msts` and values for the resource must be exactly as shown.

```
Begin Resource
RESOURCENAME TYPE INTERVAL INCREASING DESCRIPTION
...
msts Boolean () () (Windows Terminal Server)
...
End Resource
```

2. Add the msts resource to hosts.

Edit `LSF_CONFDIR\lsf.cluster.cluster_name` and add the `msts` resource to each host on which Terminal Server is installed. For example:

```
Begin Host
HOSTNAME model type server rlm pg tmp RESOURCES
...
hostA ! NTX86 1 - - - (msts)
...
End Host
```

3. (Optional) Create job starters to preserve a user's environment settings.

You may need to create a job starter script to preserve a user's environment settings on the execution host.



## Submit Platform LSF jobs to Terminal Services hosts (tssub)

Terminal Services Full Client Windows Installer (MSI) package must be installed on submission hosts.

1. Submit a job to a host with Terminal Services installed by using the `tssub` command:

**`tssub myjob`**

The `bsub` option `-o out_file` is not supported for `tssub`.

Only Windows `bsub` options are supported for `tssub`. For example, you cannot use the options `-Ip`, `-Is`, `-L login_shell` of `bsub` with `tssub`.

2. View job output:

`tspeek job_ID`

You can use `tspeek` from any LSF Windows to view the output of a Terminal Services job:

You can also use `tspeek` to monitor job output from a Linux host with `rdesktop` installed. You cannot use `tspeek` to monitor job output from UNIX.

3. Monitor the job:

**`bjobs -l`**

If you use to monitor the job, you see a message:

External Message 2 was posted from LSF\lsfadmin to message box 2

The body of the message contains the ID of the terminal session that was created.

## Limit the number of Terminal Services jobs on a host

The `msts` resource indicates to LSF whether an execution host has Terminal Server installed or not.

To limit the number of Terminal Services jobs that run on a host and keep track of how many jobs are running, define a numeric resource in addition to the `msts` boolean resource. You can alternatively use an `elim` to report how many terminal servers are available for each host.

1. Configure a numeric Terminal Server resource.

Define the resource in `LSF_CONFDIR/lsf.shared`. For example:

```
Begin Resource
RESOURCENAME TYPE INTERVAL INCREASING DESCRIPTION
...
term_server Numeric 60 N (Terminal Server)
...
End Resource
```

2. Submit a job with `rusage`.

When submitting a job, use the `rusage` resource requirement string:

```
tssub -R"rusage[term_server=1]" myjob
```

## Submit a Terminal Services job from UNIX

In mixed cluster environments, it is possible to submit a Terminal Services job with `bsub` from a UNIX host. You can use `tspeak` to monitor job output from a Linux host with `rdesktop` installed. You cannot use `tspeak` to monitor job output from UNIX.

1. On the UNIX submission host, define the environment variables `LSF_LOGON_DESKTOP=1` and `LSB_TSJOB=1`.
  - When `LSF_LOGON_DESKTOP=1`, it allows GUIs to be displayed on the execution host.
  - When the `LSB_TSJOB` variable is defined to any value, it indicates the job is a Terminal Services job.
2. Submit the job with `bsub` and indicate the `mst.s` resource requirement. For example:

```
bsub -R "msts" myjob
```



## Installing Platform LSF in a Mixed Cluster

## Set up a Linux cluster with Windows compute nodes

Complete the following steps to set up a Linux cluster with Windows compute nodes.

1. Install the Linux cluster on a shared file system.
2. Configure the Linux cluster.
3. Restart the LSF cluster.
4. Install the Windows compute node.
5. Start the Windows compute node.

### Install the Linux cluster on a shared file system

1. Install the Linux cluster as described in the Linux install guide, with exceptions to allow for Windows compute nodes.

Edit `install.conf` and specify the following:

- a) Enable dynamic hosts.

Enable or add the following line:

```
ENABLE_DYNAMIC_HOSTS=Y
```

- b) Optional. Allow EGO to control the LSF daemons.

Enable or add the following line:

```
EGO_DAEMON_CONTROL=Y
```

- c) Specify the cluster administrator.

```
LSF_ADMIN=user_account
```

For example:

```
LSF_ADMIN=lsfadmin
```

- d) Specify the installation directory.

```
LSF_TOP=directory
```

For example:

```
LSF_TOP=$SHARE/LSF_8.0
```

### Configure the Linux cluster

1. Optional. If you allowed EGO to control the LSF daemons, add Windows compute node information to the LSF service configuration files.

- a) Edit `LSF_TOP/conf/ego/cluster_name/eservice/esc/conf/services/res.xml`

- b) Navigate to the section with the correct Windows host type.

For 64-bit compute hosts, navigate to `NTX64`. For IA hosts, navigate to `NTIA64`. For other Windows compute hosts, navigate to `NTX86`.

- c) Add the proper Windows compute node information for the `Command` and `ExecutionUser` tags.

For example:

```
<sc:ActivityDescription>
```

```
<ego:Attribute name="hostType" type="xsd:string">NTX86</ego:Attribute>
<ego:ActivitySpecification>
<ego:Command>C:\LSF_8.0\8.0\etc\res.exe -3</ego:Command>
<ego:ExecutionUser>LSF\lsfadmin</ego:ExecutionUser>
<ego:EnvironmentVariable name="LSF_ENVDIR">C:\LSF_8.0\conf</ego:EnvironmentVariable>
...
```

- d) Edit LSF\_TOP/conf/ego/*cluster\_name*/eservice/esc/conf/services/sbatchd.xml
- e) Navigate to the section with the correct Windows host type.

For 64-bit compute hosts, navigate to NTX64. For IA hosts, navigate to NTI A64. For other Windows compute hosts, navigate to NTX86.

- f) Add the proper Windows compute node information for the Command and ExecutionUser tags.

For example:

```
<sc:ActivityDescription>
<ego:Attribute name="hostType" type="xsd:string">NTX86</ego:Attribute>
<ego:ActivitySpecification>
<ego:Command>C:\LSF_8.0\8.0\etc\sbatchd.exe -3</ego:Command>
<ego:ExecutionUser>LSF\lsfadmin</ego:ExecutionUser>
...
```

2. Add the Windows cluster administrator account to the your cluster file.
  - a) Edit LSF\_CONFDIR/lsf.cluster.*cluster\_name*
  - b) In the ClusterAdmins section, add LSF\lsfadmin to the Administrators list.

For example:

```
Begin ClusterAdmins
Administrators = lsfadmin LSF\lsfadmin
End ClusterAdmins
```

3. Add the LSF user domain to the lsf.conf file.
  - a) Edit LSF\_TOP/LSF8.0/conf/lsf.conf
  - b) Add the LSF user domain.

```
LSF_USER_DOMAIN=lsf_user_domain
```

For example:

```
LSF_USER_DOMAIN=LSF
```

4. Register the Windows execution password to your Linux cluster.

- a) If not using EGO to control the LSF daemons:
  - Register Windows user passwords to your cluster password file for all users submitting jobs to LSF.

```
lspasswd -u "domain\admin" -p password
```

For example:

```
lspasswd -u "LSF\lsfadmin" -p lsfpaswd
```

Passwords must be 23 characters or less.

- b) If using EGO to control the LSF daemons:
  - Log on to any host in the cluster as egoadmin.
  - Log on to EGO as the cluster administrator.

For example:

```
egosh user logon -u Admin -x mypasswd
```

- Register Windows user passwords to your cluster password file for all users submitting jobs to LSF.

```
egosh ego execpasswd -u "domain\admin" -x password -noverify
```

For example:

```
egosh ego execpasswd -u "LSF\lsfadmin" -x lsfpaswd -noverify
```

The password must be 23 characters or less.

The -noverify option is required since only a Windows host can verify the password for a Windows user.

## Restart the Platform LSF cluster

1. Restart the LSF cluster.

```
lsfstartup
```

## Install the Windows compute node

1. Install the Windows compute node as described in the Windows install guide, with exceptions to be part of a Linux cluster.

Specify the following options during installation:

- a) Specify the Linux master host as your master host name.

```
Master_Name=linux_master_name
```

- b) If you allowed EGO to control the LSF daemons in your cluster master host, allow EGO to control the LSF daemons in your compute node.

```
EGO_DAEMON_CONTROL=Y
```

- c) Specify the same port number as that of the Linux master host.

```
Port_Number=base_port_number
```

- d) Specify the cluster administrator to be the same as on your master host.

```
LSF_ADMIN=domain\user_account
```

For example:

```
LSF_ADMIN=LSF\lsfadmin
```

- e) Specify the installation directory.

```
LSF_TOP=directory
```

For example:

```
LSF_TOP=C:\LSF_8.0
```

2. Register the Windows execution user password to your cluster password file.

```
lspasswd -u "domain\admin" -p password
```

For example:

```
lspasswd -u "LSF\lsfadmin" -p lsfpaswd
```

The password must be 23 characters or less.

## Start the Windows compute node

1. Start the Windows compute node:

```
lsadmin limstartup
```



2. If you did not allow EGO to control the LSF daemons, manually start the LSF services:

**lsfadmin resstartup**

**badmin hstartup**

# Set up a Windows cluster with Linux compute nodes

Complete the following steps to set up a Windows cluster with Linux compute nodes if Ego is not controlling the LSF daemons.

1. Install the Windows cluster master host as described in the Windows installation guide.
  - a) Add the LSF user domain to the `lsf.conf` file.
    - Edit `C:\LSF8.0\conf\lsf.conf`
    - Add the LSF user domain.

```
LSF_USER_DOMAIN=lsf_user_domain
```

For example:

```
LSF_USER_DOMAIN=LSF
```
  - b) Restart the Windows cluster.

**lsfrestart**
  - c) Register Windows execution user passwords to your cluster password file for all users submitting jobs to LSF.

**lspasswd -u "domain\admin" -p password**

For example:

**lspasswd -u "LSF\lsfadmin" -p lsfpaswd**

Passwords must be 23 characters or less.
2. Install the Linux compute nodes as described in the Linux installation guide.
  - a) Specify the Windows master host as your master host name.

```
Master_Name=windows_master_name
```
  - b) Start the Linux compute nodes

# Set up a Windows cluster with Linux compute nodes and EGO controlling Platform LSF daemons

Complete the following steps to set up a Windows cluster with Linux compute nodes and EGO controlling the LSF daemons.

1. Install the Windows cluster master host as described in the Windows installation guide and allow EGO to control the LSF daemons.
  - a) Specify the following option:
 

```
EGO_DAEMON_CONTROL=Y
```
  - b) Add the LSF user domain to the `lsf.conf` file.
    - Edit `C:\LSF8.0\conf\lsf.conf`
    - Add the LSF user domain.
 

```
LSF_USER_DOMAIN=lsf_user_domain
```

For example:

```
LSF_USER_DOMAIN=LSF
```
2. Install the Linux compute node as described in the Linux installation guide and allow EGO to control the LSF daemons.
 

Edit `install.conf` and add the following line:

```
EGO_DAEMON_CONTROL=Y
```
3. Restart the Windows cluster.
 

```
egosh ego restart
```



---

# Index

## A

augmentstarter job starter 37

## C

cluster

checking 7

verifying configuration 7

COMPSPPEC environment variable 36

COMPUTERNAME environment variable 36

cross-platform user mapping 28

## D

default user mapping 28

lspasswd command 31

wgpasswd command 31

DHCP and LSF 45

DNS (Domain Name System) 46

Domain Name System 46

domain user accounts

multiple 28

drive letter format (UNC) 36

Dynamic Host Configuration Protocol 45

## E

environment variables 36

LSF\_LOGON\_DESKTOP 50

LSF\_TS\_LOGON\_TIME 50

Terminal Services jobs 50

environment, job execution 36

## F

files

license.dat

location 18

FlexNet

installs.exe 22

license manager service 22

## H

hosts

dynamic IP addressing 45

## I

installation

LSF Monitor 42

testing 5

installs.exe 22

intervals

configuring in LSF Monitor 43

## J

job execution environment 36

job starters

augmentstarter 37

preservestarter 37

## L

license file (license.dat)

TS\_OK keyword 52

license manager service 22

license server

checking status (lmstat) 24

displaying status (lmstat) 24

license.dat file

description 18

location 18

TS\_OK keyword 52

licenses

license.dat file 18

Terminal Services 52

TS\_OK keyword 52

- lstat command
  - using 24
- load information
  - charting with Windows Performance Monitor 39
- LSF configuration, checking 7
- LSF license
  - FlexNet license manager service 22
- LSF license file (license.dat)
  - description 18
  - location 18
- LSF Monitor
  - configuring
    - sample intervals 43
  - installing 42
  - starting and stopping 44
  - statistics 44
- LSF products, adding 20
- LSF\_BINDIR 36
- LSF\_DHCP\_ENV in lsf.conf 47
- LSF\_LOGON\_DESKTOP
  - variable 50
- LSF\_TS\_LOGON\_TIME
  - variable 50
- lsf.conf file
  - LSF\_BINDIR 36
- lsf.conf, LSF\_DHCP\_ENV 47
- lspasswd command
  - checking RES 9
  - default user mapping 31
  - mixed cluster configuration 63, 64, 66
  - registering user names and passwords 31
  - syntax for user names 33

## M

- mixed cluster
  - monitoring Windows Terminal Services jobs (tspeek) 59
  - starting 7
  - submitting Windows Terminal Services jobs 59
  - testing 5
  - user mapping 28
- mixed cluster configuration
  - lspasswd command 63, 64, 66
- multiple domains 28

## N

- NTRESKIT environment variable 36

## O

- OS2LIBPATH environment variable 36

## P

- passwords
  - registering with lspasswd 31
- PATH environment variable 36
- permanent LSF license
  - displaying server status (lstat) 24
- preservestarter job starter 37
- PROCESSOR\_ARCHITECTURE environment variable 36
- PROCESSOR\_LEVEL environment variable 36

## Q

- queues
  - displaying 13

## R

- Remote Execution Server (RES)
  - checking 9
- RES (Remote Execution Server)
  - checking 9

## S

- sample intervals
  - configuring in LSF Monitor 43
- servers
  - WINS, DNS, DHCP 47
- statistics in LSF Monitor 44
- syntax of user names 30
- SYSTEMDRIVE environment variable 36
- SYSTEMROOT environment variable 36

## T

- Terminal Services client
  - TS\_OK keyword 52
- Terminal Services jobs
  - environment variables 50
- TS\_OK keyword 52
- tspeek command
  - description 57
  - on Linux 59

## U

UNC file path format 36

UNIX compatibility 36

user mapping 28

user name syntax

    LSF commands 33

    LSF files 33

user names

    registering with lspasswd 31

users

    specifying 30

    viewing 30

## W

wgpasswd command

    default user mapping 31

    registering user names and passwords 31

WINDIR environment variable 36

Windows Internet Naming Service (WINS) 46

Windows Performance Monitor

    charting LSF load information 39

    configuring LSF Monitor 43

    installing LSF Monitor 42

WINS (Windows Internet Naming Service) 46