



# **Grid MP™ Platform Version 4.1**

**Application User's Quick Guide**



# Copyright, Trademark, and Version Notices

## Copyright Notice

Copyright © 2001-2004 United Devices, Inc. All rights reserved.

Information relating to the architecture and software described in this guide is proprietary to United Devices and, as applicable, its licensors. Rights to use the software and the accompanying documentation are governed by a separate license agreement from United Devices presented prior to software installation.

Information in this document is subject to change without notice. No part of this guide may be reproduced or published in any form or by any means, electronic or mechanical, including photocopying, for any purpose other than the purchaser's personal use.

## Trademark Notice

United Devices, UD, Grid MP, and MP Agent are trademarks or service marks of United Devices, Inc. Other company and product names may be the trademarks or service marks of third parties. Third party software is installed as distinct, unmodified software programs during the Grid MP™ platform installation process. Licensing information pertaining to those third party programs is available on the Grid MP Installation CD-ROM in `/mnt/cdrom/licenses.html`.

## Document and Software Versions

- Document Version 2
- Software Version 4.1



# Contents

Audience .....	vii
Related documentation .....	vii
Using this guide .....	viii
Guide organization .....	viii
Typographical conventions .....	ix
Viewing this guide online .....	ix
Printing this guide .....	ix
Software maintenance and upgrades .....	x
Contacting United Devices .....	x
<b>Chapter 1: Introduction to Grid MP Job Submission and Retrieval .....</b>	<b>1</b>
What Is the Grid MP Platform? .....	1
How Do I Access the Grid MP Platform? .....	1
Command Line .....	2
Grid MP Console .....	2
Application-specific Command-line Interfaces or Consoles .....	3
What Do I Need to Get Started? .....	3
How Are Jobs Scheduled on the Grid MP Platform? .....	3
<b>Chapter 2: Command-line Job Submission .....</b>	<b>7</b>
Submitting a Batch Job .....	7
Batch Prerequisites .....	7
Usage .....	8
mpsub Example .....	11
Submitting an MPI Job .....	11
Submitting MPI Jobs with ud_mpirun .....	12
MPI Prerequisites .....	12
Usage .....	13
ud_mpirun Example .....	15
Submitting Third-Party MPI Jobs with mpsub .....	16
Retrieving Results for Batch and MPI Jobs .....	17
Usage .....	17
mpresult Example .....	19

---

Deleting Jobs .....	19
Viewing Job Status .....	19
<b>Chapter 3: Grid MP Console Job Submission .....</b>	<b>21</b>
Logging on to the Grid MP Console .....	21
Submitting a Job to Run on Devices .....	22
Creating a Job and Job Step in the MP Console .....	22
Checking Job Status .....	25
Retrieving Results and Errors .....	26
Retrieving Results in the MP Console .....	26
Retrieving Errors in the MP Console .....	27
Deleting Results and Jobs .....	27
Deleting a Job in the MP Console .....	27
<b>Appendix A: Application Administrator Tasks .....</b>	<b>29</b>
mpbatch and ud_mpirun Prerequisites .....	29
All Submission Machines .....	29
Initializing the mpbatch and ud_mpirun Utilities .....	30
On Windows .....	31
On Linux .....	31
UD_MPIRUN .....	32
Rebuilding the MPICH Linux Library .....	32
Relinking the MPICH Linux Library .....	33
Preparing the Grid MP Platform to Run Third-Party MPI Jobs .....	33
Creating Data Packages with buildpkg .....	34
Digital Signatures and MPBATCH and UD_MPIRUN .....	35

## About This Guide

The United Devices® Grid MP™ platform is a distributed computing system that enables your organization to coordinate and share existing computing, application, data, storage, and network resources across departmental and geographically dispersed locations to form a high-performance, cost-effective application processing platform.

The *Grid MP platform Version 4.1 Application User's Quick Guide* describes how to perform the following tasks:

- Create Jobs in the Grid MP Console
- Use command-line utilities to submit batch and MPI Jobs
- Retrieve result and error information from the command line and the Grid MP Console

In addition, this guide provides an overview of the interfaces to the Grid MP platform what you need to get started.

## Audience

The audience for this guide includes Job submitters who are new or returning to the Grid MP platform, and anyone who wants to get a quick overview of how to complete common tasks on the platform.

## Related documentation

The following United Devices documentation is also useful to Grid MP platform users:

- *Grid MP platform Version 4.1 Application Developer's Guide*, which describes how to develop or migrate applications using the Grid MP Software Developer's Kit.

- *Grid MP platform Version 4.1 Installation Guide*, which provides step-by-step instructions for planning and installing the Grid MP platform.
- *Grid MP platform System Administrator's Guide*, which contains help for using the Grid MP platform to manage Jobs and Data, as well as system configuration information.
- MP Console online help, which provides context-sensitive online help for MP administration tasks. To access the context-sensitive online help from any MP Console window, click the **Help** link in the top right corner of the MP Console screen.
- United Devices Web site at <http://www.ud.com/>, which provides general information about the Grid MP platform.

In addition, Grid MP platform Application Users need documentation for any third-party application services they will use to submit Jobs and retrieve Results.

## Using this guide

This section describes how this guide is organized, its typographical conventions, and provides information about printing the guide.

### Guide organization

This *System Administrator's Guide* contains the following chapters:

- Chapter 1, “[Introduction to Grid MP Job Submission and Retrieval](#),” on page 1 introduces the Grid MP platform interfaces, outlines what you need to get started, and describes how jobs are scheduled.
- Chapter 2, “[Command-line Job Submission](#),” on page 7, explains the batch and MPI command-line interfaces to the Grid MP platform.
- Chapter 3, “[Grid MP Console Job Submission](#),” on page 21, describes how to use the Grid MP Console to submit Jobs and retrieve Results.
- Appendix A, “[Application Administrator Tasks](#),” on page 29, describes the tasks an Application Administrator will need to do before you can use the batch and MPI utilities to submit Jobs and provides information about creating Data packages.
- “[Application User's Quick Guide Feedback Form](#)” on page 37, is a form you can use to provide feedback about this Guide. Print this form, write your comments, and fax it to us at +1-512-331-6235.



## Typographical conventions

The following table shows the typographical conventions used in this book.

**Table 1: Typographical conventions**

<b>Convention</b>	<b>Description</b>
<code>Courier</code>	Code examples and screen output
<b>Courier</b>	Functions, commands, and user input
<i>Italic</i>	Function and command place holders—replace with the appropriate name or value, such as a file name or path name.
<b>Bold</b>	Menu names, menu commands, menu options, and buttons

## Viewing this guide online

This document is best viewed with Adobe<sup>®</sup> Acrobat<sup>®</sup> Reader<sup>®</sup> version 5.0. If you don't have this version of the Acrobat Reader, you can download it here at no charge: <http://www.adobe.com/products/acrobat/readstep2.html>.

This document contains a number of features that will enable you to navigate quickly and easily through the document:

- Bookmarks are on the left side of the screen. To use the bookmarks, click the bookmark for that section of the document that you want to view. To return to the previous view, right-click and select **Go To Previous View** from the pop-up menu.
- The index is at the end of this document. To go to the page for the topic you want to view, click the page number beside the topic. To return to the previous view, right-click and select **Go To Previous View** from the pop-up menu.
- Links to figures and sections can be seen throughout the document. These links allow you to go to a figure, table, or section that is referenced in the text. You can click any of these links to view the referenced figure, table, or section. To return to the previous view, right-click and select **Go To Previous View** from the pop-up menu.

## Printing this guide

This guide is formatted for 8.5" x 11" size paper and designed for duplex printing. A printer that supports duplex printing will print one page of this book on each side of a sheet of paper.

## Software maintenance and upgrades

United Devices is committed to continuously improving its products and services. When new maintenance or product releases are available, United Devices Professional Services will contact you to help plan your upgrade. Upgrades involve assessing application impact, preparing database migration scripts, and testing upgrade procedures. Some licensing issues may also apply. For more information about upgrades, contact United Devices Professional Services.

## Contacting United Devices

For technical support or to inquire about the information contained in this document, contact United Devices in one of the following ways:

- For technical support, send e-mail to [customer@ud.com](mailto:customer@ud.com).
- For technical support, telephone (800) 370-5320 between 9 A.M. and 5 P.M. (CST) Monday through Friday.
- For Grid MP platform consulting or training, send e-mail to [customer@ud.com](mailto:customer@ud.com), or see <http://service.ud.com/>.

# Chapter 1: Introduction to Grid MP Job Submission and Retrieval

## What Is the Grid MP Platform?

The United Devices<sup>®</sup> Grid MP<sup>™</sup> platform is a distributed computing system that enables your organization to coordinate and share existing computing, application, data, storage, and network resources across departmental and geographically dispersed locations to form a high-performance, cost-effective application processing platform.

As the Grid MP platform user, you can utilize the high-performance distributed computing system to run your executables against data you provide to solve problems such as large-scale research projects, product design processes, or other processing-intensive computation tasks. The following types of users exist in the Grid MP platform:

- **Application User**—Uploads executables and data to the Grid MP platform; creates Jobs and submits them to run on the MP Agent machines; monitors and downloads results.
- **Application Developer**—Migrates, develops, and deploys applications for the Grid MP platform. Creates Application Services that perform application preprocessing and postprocessing for use by the Application User.
- **System and/or Application Administrator**—Installs, configures, and maintains the Grid MP platform software. Manages user security, service performance, and Device usage.

This document is primarily intended to enable the Application User to get started using the Grid MP Platform without much preparation or training.

## How Do I Access the Grid MP Platform?

United Devices provides the following interfaces to allow you to perform work with the Grid MP platform. Ask your System Administrator or Application Administrator to help you determine which interface(s) your organization uses.

---

## **Command Line**

The Grid MP platform supports the command-line submission of batch and MPI Jobs, and their result retrieval using the mpinit, mpsub, ud\_mpirun, and mpresult command-line utilities. In-depth instructions for how to submit Jobs and retrieve their Results using the utilities is available in Chapter 2: “[Command-line Job Submission](#),” on page 7.

### **mpinit**

The mpinit utility initializes batch Jobs and ud\_mpirun jobs. mpinit must run once prior to running the first batch Job using mpsub. In most cases, your System or Application Administrator will complete this step for you.

### **mpsub**

The mpsub batch submission utility is a method to submit both the executable and data in a single unit of work. It is intended for use with computational requirements that may change frequently. Application users run mpsub by submitting all the Programs for the Application and its input data to the Grid MP platform as a Job.

### **ud\_mpirun**

ud\_mpirun enables you to run MPICH-based parallel Jobs to run on the Grid MP platform. The interface enables any arbitrary MPICH executable to run on supported operating system platforms. When using ud\_mpirun, you can specify input at the command line or specify a configuration file that contains ud\_mpirun input.

### **mpresult**

The mpresult utility retrieves the Results produced when you use mpsub to run a batch Job or ud\_mpirun to run an MPI Job. You can also use mpresult to view a list of all running mpsub and ud\_mpirun Jobs and to stop or delete the Jobs.

## **Grid MP Console**

The Grid MP Console is a Web-based client for submitting Jobs to run on MP Agents. The Grid MP Console is best used for large Jobs with multiple Workunits. The Grid MP Console includes context-sensitive help for each HTML screen and step-by-step instructions for commonly performed tasks.

United Devices supports the following browsers for the MP Console:

- Microsoft Internet Explorer 5.5 (Service Pack 1) or 6.0
- Netscape<sup>®</sup> Navigator 6.2.3 or 7.0
- Mozilla 1.01.+

## ***Application-specific Command-line Interfaces or Consoles***

In some cases you will use neither the generic command-line interfaces or the Grid MP Console to submit jobs and retrieve results. The MP Grid Services Interface (MGSI) is a programmatic interface that provides third-party applications with a way to access to the Grid MP Services and Database. As an application user, you usually do not interact directly with the MGSI. You may, however, utilize a third-party application service written using MGSI calls as your primary user interface. As this type of interface is specific to a single application, instructions for using it are not provided within this document. Contact your System or Application Administrator for more information.

## **What Do I Need to Get Started?**

The Grid MP platform model is designed to support a variety of high-performance computing applications, including coarse-grained, data-parallel Jobs, sequential batch Jobs and more tightly coupled parallel jobs that use MPI or similar parallel programming environments. Depending on what you want to accomplish with the Grid MP platform, at the very least you will need the following:

- Grid MP user name and password
- Membership in a Grid MP User Group with roles and privileges that allow you to perform the necessary actions and access the necessary data to get your work done. The following chapters contain step-by-step instructions for using the Grid MP platform to run jobs and collect results. The roles and privileges required for each task are specified as part of the instructions.

Your System or Application Administrator can provide you with both of the above requirements. You will also need to know which user interface your organization expects you to employ to access the Grid MP platform.

Depending on the interface you use to run your Jobs, you will need other things, such as executables and input files. The following chapters contain in-depth instructions, including prerequisites, for generic command-line and MP Console Job submission.

## **How Are Jobs Scheduled on the Grid MP Platform?**

When a Job is submitted to the Grid MP Platform its workload is first split into Workunits. Once that is complete the Grid MP platform Scheduler sends Workunits to Devices using the Grid MP platform dispatch algorithm.

1. The following factors are taken into consideration to create a list of qualified Jobs:
  - Workunit Timeout—Job Creators can set the maximum amount of time a Workunit can take before it is aborted. Device Managers can also specify maximum Workunit timeouts for Device Groups. The workload scheduler

---

will only send Jobs to Device Groups with a maximum Workunit timeout equal to or greater than the Job's maximum Workunit timeout.

- Workload Thresholds—Job Managers can specify the following properties for a Job to ensure results are dependable:
    - Number of Results—The minimum number of Results a Workunit must have in order to be considered complete.
    - Maximum Number of Errors—The maximum number of Errors a Workunit can received before it fails.
    - Maximum Concurrent Workunits Dispatched—The maximum number of times any one Workunit can be dispatched at once.
  - Resource Constraints—When a Job is executed it runs a previously installed Program on the Grid MP platform. Application administrators set specific resource requirements when a Program is installed, such as the minimum free disk space, CPU clockrate, or physical RAM a Device must have to run the Program. Device Group Managers specify maximum utilization thresholds for their Devices, which affects which Jobs can run on the Devices.
  - Device Preferences—Device Group Managers set the Programs a Device or Device Group can run with a Device Profile.
  - Device Group Targeting—Job creators can target specific Device Groups for their Jobs. If a Job is targeted to Device Groups, it will only run on those Device Groups. If a Job has no targeted Device Groups, it can run on any Device Group that meets its resource constraints.
2. From the list of qualified Jobs, user scheduling priorities at the Device Group level are compared to select a user. This priority setting determines the service rate of all Jobs for a user or user group as compared to other users or user groups submitting Jobs to the Device Group in question. Device Group administrators set this priority level.
  3. From the Jobs submitted by the selected user, Job priorities are compared to select a Job. This priority allows the Job creator to prioritize among their Jobs. Job Managers set this priority level when creating a Job in the console. Priority can also be set for mpsub and ud\_mpirun Jobs.
  4. Optimal Workunit Selection is applied to find the best Workunit for the selected Job. The scheduler attempts to give higher preference to a Workunit that uses a Data file the MP Agent on the Device has previously downloaded. This reduces the amount of network traffic used when downloading the files for the Workunit, since the MP Agent may have cached some of them.

Using all these constraints a Job and its associated Workunits will be scheduled to run on the target set of Devices at the appropriate priority. Once a Job is running, a user with the necessary roles and privileges may alter various runtime parameters, such as

the Job's priority, scheduled end time, timeouts, minimum results and maximum errors.

**NOTE** When you submit an MPI job, the Grid MP Scheduler looks for Device Groups that can run the MPI job and starts sending Workunits out to those Agents requesting work from those Device Groups. Since MPI workunits are gang-scheduled, the Scheduler will not send Workunits from other jobs to those Agents until all the MPI-job workunits are scheduled. Targeting MPI jobs to Device Groups associated with a cluster is recommended.

Contact your system administrator for help adjusting the rate at which your Jobs are scheduled.

---



## Chapter 2: Command-line Job Submission

This chapter describes the command-line submission utilities, `mbsub` and `ud_mpirun`, and the result management utility, `mpresult`.

**NOTE** All command line utilities are provided in platform specific versions. The perl source code of all the command line utilities is provided in the Grid MP SDK.

### Submitting a Batch Job

The batch submission facility allows Job submitters to submit both the executable and associated Data in a single step. The `mbsub` utility is meant for small Jobs that require only one unit of work. Note that `mbsub` uses applications directly, i.e. that have not been pre-processed using the `buildmodule` tool.

### Batch Prerequisites

The following instructions assume that you or your System Administrator have already copied `mbsub` to the location from where you want to run it, properly set the `uduserconf` file accordingly, and executed the `mpinit` script. See “[All Submission Machines](#)” on page 29 for installation instructions if this is not the case.

To use `mbsub.pl` you need perl libraries and dependencies to be installed on your system. Installation details about perl is explained in detail in the ‘Satisfying Prerequisites’ section in the *Application Developer’s Guide*.

Alternatively, you can use the perl executables to run the `mbsub` script. These executables do not require perl libraries to be installed on your machine. The following list identifies the platform-specific `mbsub` executables:

- **`mbsub.exe`**—for Windows
- **`mbsub_i686-pc-linux-gnu`**—for Linux

- **mpsub\_powerpc-ibm-aix5.2.0.0**—for AIX
- **mpsub\_sparc-sun-solaris2.8**—for Solaris

To run the `mpsub.pl` script a `uduserconf` file must reside in the home directory.

To submit a batch Job, you need an executable and possibly one or more input files. The user submitting the Job must have Create Job, Update Application, and Read Application Privileges on the “batch” application, which is created by the `mpinit` script, the Program Creator Role, and the Read Device Group privilege.

## Usage

To submit a batch job, use the following command line syntax:

```
mpsub.pl [OPTIONS] program commandline
```

Where:

- **mpsub.pl** is `mpsub` script.  
You can also use the following pared versions to run `mpsub`. These executables do not require a perl environment.
  - **mpsub.exe**—For Windows
  - **mpsub\_i686-pc-linux-gnu**—For Linux
  - **mpsub\_powerpc-ibm-aix5.2.0.0**—For AIX
  - **mpsub\_sparc-sun-solaris2.8**—For Solaris

<b>NOTE</b>	The platform-specific utility binary versions may not work on some AIX or Solaris computers. If the PAR executables do not work, run the Perl (*.pl) version of the script. Before running the perl version of the script, ensure that the dependent Perl CPAN modules are installed on the machine where you plan to run the script.
-------------	---

- **OPTIONS** are command-line arguments specific to the MGSI, the Program Module executable, the batch Job, or the `mpsub` utility. All available options are described in [Table 1](#) below.

Some options accept elective values, indicated by square brackets in the descriptions in [Table 1](#). If you declare an option that accepts an elective value but you do not supply a value, the *program* may be mistaken for the value of the option if there is not another option following it. To avoid this, either specify the value of the option explicitly, or follow the option with the missing value by another option.

- *program* is the path to the executable you want to run.
- *commandline* refers to any arguments you pass to the executable.

After submitting a batch Job, a unique id number representing the Job will be returned. Keep the id number for use in retrieving results.

**Table 1: mpsub Option Descriptions**

Option	Description
<b>MGSI Options</b>	
-username <i>name</i> or -user <i>name</i>	The username to use when logging on to the Grid MP system.
-password <i>password</i>	The password to use when logging on to the Grid MP system.
-config <i>filename</i>	The file name from which to load MGSI defaults. The default value is <code>~/uduserconf</code>
<b>Program Module executable options</b>	
-additional <i>file</i>	An additional executable file to include in the batch. This is intended to include files such as DLLs.
-define <i>name</i> [= <i>value</i> ]	An environment variable <i>name</i> to be passed to the batch Job. If the '=' sign is included, then <i>value</i> will be the environment variable value. If the '=' sign is not included, the current value of the variable from the shell environment is used.
-workunit-compression type	The type of compression to be used for sending Workunit files to a Device. Compression type options include "bzip2" and "gzip". By default the Workunit files are "gzip" compressed. If you do not want to compress the files, use the "none" option.
-result-compression type	The type of compression to be used for Result files returned by a Device. By default the Result files are "gzip" compressed. If you do not want to compress the files, use the "none" option.
-input <i>input</i> [= <i>internal</i> ]	An input file on the local filesystem to be submitted with the batch. When executed on an MP Agent, the program will be able to open the file by that name. If <i>input</i> is of the form "external= <i>internal</i> ", then the file named "external" is copied from the local filesystem. When executed on an MP Agent, the program will be able to open the file by the name "internal". The internal name may be "stdin", in which case the input file is supplied as standard input to the executable. This option may appear zero or more times to specify multiple input files.

**Table 1: mpsub Option Descriptions (continued)**

Option	Description
-output <i>output</i>	<p>After executing the program on an MP Agent, any file by this name will be copied from the MP Agent working directory to the output file.</p> <p>The file name may be "stdout" or "stderr", in which case the standard output or standard error stream will be copied to the output file.</p> <p>This option may appear zero or more times to specify multiple output files.</p>
<b>Batch Job options</b>	
-platform <i>platform</i>	<p>The platform on which the Job can run. If not specified, the platform defaults to the platform from which mpsub is executed.</p> <p>The script accepts the following platform strings:</p> <ul style="list-style-type: none"> <li>i686-pc-linux-gnu</li> <li>i686-pc-win32-9x</li> <li>i686-pc-win32-nt</li> <li>powerpc-ibm-aix5.2.0.0</li> <li>sparc-sun-solaris</li> </ul>
-cputimeout <i>seconds</i>	The number of cpu time seconds before the Job should time out.
-endtime <i>time</i>	The Job end time in unix seconds.
-priority <i>priority</i>	The priority of the Job relative to other Jobs submitted by the user. 1=lowest priority; 100=highest priority. The default is 10.
-numresults <i>num</i>	The number of redundant results per Job.
-start <i>time</i>	The Job start time in unix seconds.
-wallclocktimeout <i>seconds</i>	The number of wall clock seconds before the Job should time out.
-dgfile <i>filename</i>	The name of the file, which contains the Device Group names or gids. The file should list the Device Group name or gid per line.
<b>Other mpsub options</b>	
-block	Indicates that mpsub should wait for the Job to complete and a result to be returned before exiting.
-blockwait <i>seconds</i>	Indicates the time mpsub should wait for result retrieval.
-load [ <i>filename</i> ]	Indicates that mpsub should load default options from a <i>filename</i> that was previously created with the -save option. If <i>filename</i> is not specified the default file is mpsub.opt.
-save [ <i>filename</i> ]	Indicates that mpsub should have default options to <i>filename</i> . Saved files can be used with -load to load a default option set. If <i>filename</i> is not specified, this option saves to the mpsub.opt file. This option will not overwrite an existing file, so to update saved options, delete the previous option file first.

### ***mbsub Example***

The following is an example of submitting work using **mbsub** and the type of output seen by the user after the command is run:

```
[user@system mpi]$ ./mbsub.pl -input textfile -output output-
file -input inclusionsfile oic textfile outputfile exclusions-
file
Initializing batch program for platform i686-pc-linux-gnu
Found batch program for i686-pc-linux-gnu (B7C8C062-695A-11D8-
97C3-0090279D6EA5)
Initializing batch programversion
Found batch programversion (B8EBFC66-695A-11D8-97C3-
0090279D6EA5)
Initializing batch programmodule
Found batch programmodule (BA2E2E1E-695A-11D8-97C3-
0090279D6EA5)
Uploading programmodule executable
Initializing programmoduleversion
Added programmoduleversion (BDFB4A9A-695A-11D8-97C3-
0090279D6EA5)
job_id: 103
```

This example uses the OIC Application to create a Job in the system.

## Submitting an MPI Job

You can submit an MPI Job using `ud_mpirun` or `mbsub`. The following table compares the utilities to assist you in deciding which to use to submit your Job:

**Table 2: MPICH vs. third-party MPI instances on the Grid MP platform**

MPI using mbsub	MPI using ud_mpirun
Uses mbsub to submit the Job	Uses ud_mpirun to submit the Job
Can use any 3rd-party MPI version	Must use MPICH 1.25
Can still run the executable if it is a commercial product and the user does not have access to the source code and cannot recompile and relink it.	Must recompile using MPICH 1.25. For Linux, you must relink with the UD version of MPICH libraries that allows gang-scheduling of Workunits.
Only the head node must have the MP Agent installed.	Grid MP Agents are installed on all nodes. Results and Errors are on a per-node basis.
You have to decide which computers to use, list them in a "host file", and be sure that they are all available.	The Dispatcher picks which of the available Devices to use as nodes based on targeted Device Groups.

**Table 2: MPICH vs. third-party MPI instances on the Grid MP platform**

MPI using mpsub	MPI using ud_mpirun
If MP Agents are installed on other nodes, administrators will need to set thresholds for the Devices to back them off if the Device resources are in use by another program so that MPI Jobs run serially.	The system monitors polling and can detect when a Device stops polling and becomes dormant. The MP Agent monitors the processes and can shut down the processes if the Job is aborted or fails early.

## Submitting MPI Jobs with ud\_mpirun

To submit MPI-based parallel Jobs to run on the Grid MP platform, you utilize a command-line script—`ud_mpirun`—instead of the MP Console. Like `mpsub`, the `ud_mpirun` utility allows you to submit the executable and any input files at the command line. `ud_mpirun` also creates Program components each time you run it, and the Grid MP platform automatically deletes the Application and components when you delete the Job.

### MPI Prerequisites

The following instructions assume you or your System Administrator have already installed MPICH version 1.2.5 for both Linux and Windows deployments, properly set the `uduserconf` file, and executed the `mpinit` script. See “[All Submission Machines](#)” on page 29 for installation instructions if this is not the case.

To use `ud_mpirun.pl` you need perl libraries and dependencies to be installed on your system. Installation details about perl is explained in detail in the “Satisfying Prerequisites” section in the *Application Developer’s Guide*.

Alternatively, you can use the perl executables to run the `ud_mpirun` script. These executables do not require perl libraries to be installed on your machine. The following list identifies the platform-specific `ud_mpirun` executables:

- **`ud_mpirun.exe`**—for Windows
- **`ud_mpirun_i686-pc-linux-gnu`**—for Linux
- **`ud_mpirun_powerpc-ibm-aix5.2.0.0`**—for AIX
- **`ud_mpirun_sparc-sun-solaris2.8`**—for Solaris

To submit a Job with `ud_mpirun`, you need an executable and possibly one or more input files. The user submitting the Job must have Create Job, Update Application, and Read Application Privileges on the “`mpi`” application, which is created by the `mpinit` script, the Program Creator Role, and the Read Device Group privilege.

To run the `ud_mpirun` script a `uduserconf` file must reside in the home directory.

**NOTE** On Linux, you or your Administrator must also rebuild your MPICH library to include UD-specific patches and to relink your application to use the updated library before running `ud_mpirun`. On Windows there is no UD-specific step beyond normal MPI development. See the instructions in “[Rebuilding the MPICH Linux Library](#)” on page 32 for assistance with Linux deployments.

## Usage

`ud_mpirun` is located on the Grid MP CD-ROM in the `UDsdk_4.1/tools/mpi` directory.

**NOTE** Ensure that you use the `ud_mpirun` utility instead of the `mpirun` utility. Using the `mpirun` utility will cause Job submission to fail.

Use the following command-line syntax to submit an MPI Job with `ud_mpirun`

```
ud_mpirun.pl -np numprocs [options] executable_name [arguments]
```

Where:

- `ud_mpirun.pl` is the `ud_mpirun` script.  
You can also use the following pared versions to run `ud_mpirun`. These executables do not require a perl environment.

- `ud_mpirun.exe`—For Windows
- `ud_mpirun_i686-pc-linux-gnu`—For Linux
- `ud_mpirun_powerpc-ibm-aix5.2.0.0`—For AIX
- `ud_mpirun_sparc-sun-solaris2.8`—For Solaris

**NOTE** The platform-specific utility binary versions may not work on some AIX or Solaris computers. If the PAR executables do not work, run the Perl (\*.pl) version of the script. Before running the perl version of the script, ensure that the dependent Perl CPAN modules are installed on the machine where you plan to run the script.

- `numprocs` is a placeholder for the total number of processes to run and is a required argument. Other options are described in the table below.
- `options` are command-line arguments specific to `ud_mpirun` or the MPI Job.

**NOTE** To run the Job successfully on Windows, an additional file named `mpich.dll` must be associated with the Job. Type `-f mpich.dll` to specify the file.

- *executable\_name* is a placeholder for the path to the executable you want to run. You can pass arguments to the executable at the command line.
- Arguments are any command-line arguments the MPI executable takes.

After submitting the Job, a unique id number representing the Job will be returned. Keep the id number for use in retrieving results.

**Table 3: ud\_mpirun Option Descriptions**

Option	Description
<b>Login options</b>	
-username <i>username</i>	The username to use when logging on to the Grid MP system.
-password <i>password</i>	The password to use when logging on to the Grid MP system.
<b>General options</b>	
-np <i>numprocs</i>	The total number of processes to run. This is a required argument.
-f <i>file</i>	Additional files to be packaged with executable. You can specify multiple files.
-c <i>file</i>	A config file that contains <code>ud_mpirun</code> input to read instead of <code>.uduserconf</code> .
-env <i>name1=value1 name2=value2</i>	Environment variables required to run the MPI Job.
-R <i>file</i>	Utilize the -R option to provide a stdin redirection. The file you specify with a -R gets redirected to the program's stdin.
-i <i>file</i>	An input file which will be sent to every host. You can specify multiple files.
-o <i>file</i>	An output file which will be captured on every host. You can specify multiple files.
-p <i>platform</i>	The script accepts the following platform strings: i686-pc-linux-gnu i686-pc-win32-nt
-w	Indicates <code>ud_mpirun</code> should wait for the Job to complete before exiting.
<b>MPI Job options</b>	
-cputimeout <i>seconds</i>	The number of CPU time seconds before the Job should time out. No limit by default
-dgfile <i>filename</i>	The name of the file containing a list of Device Groups to target. The file should have one Device Group GUID per line.



**Table 3: ud\_mpirun Option Descriptions**

Option	Description
-endtime <i>time</i>	The Job end time in unix seconds.
-priority <i>priority</i>	The Job priority relative to other Jobs created by the user. 1=lowest priority; 100=highest priority. The default is 10.
-starttime <i>time</i>	The Job start time in Unix seconds. Defaults to current time.
-clocktimeout <i>seconds</i>	The number of wall clock seconds before the Job should time out. No limit by default.

***ud\_mpirun Example***

The following is an example of submitting work using `ud_mpirun` and the type of output seen by the user after the command is run:

```
[user@system mpi]$ ./ud_mpirun -np 16 -i textfile -o outputfile
-i exclusionsfile oic-mpi textfile outputfile exclusionsfile
Using MPI listener port 12345
Creating new MPI Program.
Creating new ProgramVersion.
Creating new ProgramModule.
Creating new ProgramModuleVersion.
Created ProgramModuleVersion: oic-mpi1077910139
Created Data/Workunit 0
Created Data/Workunit 1
Created Data/Workunit 2
Created Data/Workunit 3
Created Data/Workunit 4
Created Data/Workunit 5
Created Data/Workunit 6
Created Data/Workunit 7
Created Data/Workunit 8
Created Data/Workunit 9
Created Data/Workunit 10
Created Data/Workunit 11
Created Data/Workunit 12
Created Data/Workunit 13
Created Data/Workunit 14
Created Data/Workunit 15

Created Job 104
```

This example uses the MPI version of the OIC Application `oic-mpi` to create a Job in the system.

## Submitting Third-Party MPI Jobs with mpsub

You can use mpsub to run third-party MPI Jobs. This section describes the required steps and provides examples you can use to get started. The example given below is for running LAM MPI Jobs.

**NOTE** The instructions in this section assume that you or your System Administrator have configured the Grid MP platform to run third-party MPI Jobs. You can find step-by-step instructions for the necessary preparation in [“Preparing the Grid MP Platform to Run Third-Party MPI Jobs”](#) on page 33.

1. Create a script that boots your MPI application. Name the script ‘mywhoami’. The following example code uses oic-mpi, which comes with the Grid MP CD-ROM.

```
#!/bin/sh
whoami
lamboot ~/bhosts
chmod +x /share/mpbatch/oic-mpi
mpirun C /share/mpbatch/oic-mpi /share/mpbatch/fragment.txt
      output.txt /share/mpbatch/exclusion.txt
lamhalt
```

Where /share/mpbatch/ can be replaced by any location where oic-mpi, fragment.txt, or exclusion.txt is located. oic-mpi, fragment.txt, and exclusion.txt must be provided with their absolute path.

Add any additional application-specific setup or cleanup to the script.

Mark the script as executable before submitting. Use the following command:

```
chmod +x mywhoami
```

2. Create a file containing one line which is the name of the head node Device Group created for the MPI job. In this example, that file is named “dggroup.”

**NOTE** You or your System Administrator should have previously created a Device Group for this purpose as part of the configuration steps. You can find step-by-step instructions for the necessary preparation in [“Preparing the Grid MP Platform to Run Third-Party MPI Jobs”](#) on page 33.

3. Submit your application with `mpsub`. See “Usage” on page 8 for a list of the `mpsub` command-line options.

```
$ mpsub -dgfile dgggroup -input exclusion.txt -input
      fragment.txt -output output.txt -output stdout -output
      stderr -additional oic-mpi mywhoami
```

`mpsub` displays output such as the following:

```
Validating DeviceGroups
Targeting C35AAD36-B85F-466C-87E0-4601AE4A20CF (Head Node)
job_id: 204
```

4. Using the `job_id` number `mpsub` returns in the output, retrieve your results with `mpresult`. See “Retrieving Results for Batch and MPI Jobs” on page 17 for instructions.

## Retrieving Results for Batch and MPI Jobs

The `mpresult` script allows you to retrieve Results from Jobs you submitted using `mpsub` or `ud_mpirun`. You can also use the script to view the status of those Jobs, stop their processing, or delete them. The `mpresult` script is located in the `UDsdk_4.1/tools/mpbatch` directory.

Each result is returned as a `.tar` file containing the result file(s). In the case of `mpsub`, if the `-gzip` option was specified when the Job was submitted, the files within the result `.tar` file are also gzip compressed. There is one result file returned for each instance of the Job that was executed on an MP Agent.

### Usage

To retrieve results, you need a Job ID, and the Read Job Privilege for the Job.

```
mpresult.pl [OPTION] jobid
```

Where:

- `mpresult.pl` is the `mpresult` script.  
You can also use the following pared versions to run `mpresult`. These executables do not require a perl environment.
  - `mpresult.exe`—For Windows
  - `mpresult_i686-pc-linux-gnu`—For Linux

- `mpresult_powerpc-ibm-aix5.2.0.0`—For AIX
- `mpresult_sparc-sun-solaris2.8`—For Solaris

**NOTE** The platform-specific utility binary versions may not work on some AIX or Solaris computers. If the PAR executables do not work, run the Perl (\*.pl) version of the script. Before running the perl version of the script, ensure that the dependent Perl CPAN modules are installed on the machine where you plan to run the script.

- **OPTION** apply to all listed Jobs. `mpresult` options are described in the table below.
- When you submit a Job using `mpsub` or `ud_mpirun`, the script returns a *jobid* to you. You can specify more than one jobid on the same command line.

**Table 4: mpresult Option Descriptions**

Option	Description
-delete	Delete the indicated Job(s). When this option is specified, no results are retrieved unless -get is also specified.
-get	Get results for the specified Job(s). This is the default unless -delete is specified.
-list	List all visible Jobs.
-config <i>filename</i>	Load MGS1 defaults from a specific filename (defaults to <code>~/uduserconf</code> )
-password <i>password</i>	The password to use when logging in. This may also be specified using the environment variable <code>MP_PASSWORD</code> .
-stop	Stop the indicated Job(s).
-username <i>name</i> or -user <i>name</i>	Set the user login name to <i>name</i> . This may also be specified using the environment variable <code>MP_USER</code> .
-root	Only retrieve the results for Workunits with index zero (useful for MPI Jobs where only the root node result is desired)

### ***mpresult Example***

The following is an example of retrieving results using **mpresult** and the type of output seen by the user after the command is run:

```
[user@system mpi]$ ./mpresult -get 814
processing job 814...
result saved in result-000810-index-000000-job-000814.tar
no errors for Job 814, JobStep E71C3DE8-9750-11D8-83C6-
00E04C7772BC
```

## **Deleting Jobs**

You can use **mpresult** to delete **mpbatch** and **ud\_mpirun** Jobs. Deleting the Jobs automatically deletes Results. Use the following command to delete a Job:

```
mpresult -delete jobid
```

You can also delete a Job in the MP Console. See “[Grid MP Console Job Submission](#)” on page 21 for help using the MP Console.

## **Viewing Job Status**

You can log on to the MP Console to view information about Jobs submitted at the command line. If the Job fails, you can find information about any Errors reported as well. See “[Grid MP Console Job Submission](#)” on page 21 for help using the MP Console.



## Chapter 3: Grid MP Console Job Submission

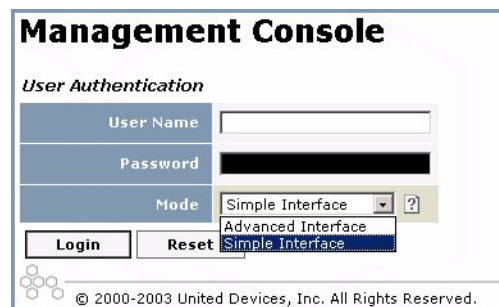
This chapter provides step-by-step instructions for using the Grid MP Console to perform common tasks, such as logging on, creating a Job, and retrieving results.

To be able to submit a non-MPI, non-batch executable, and data to run on multiple Devices, the following objects must be created in the Grid MP platform. In most cases, an Application Administrator will create the Application and Program components, and you will either submit Jobs through the Grid MP Console or through an Application Service user interface provided specifically for your Application.

- *Application*—An Application is the container for a Program or a set of Programs and the definition of the control and data flow across them.
- *Program*—A Program is a set of executable code components that perform work in the system.
- *Job*—A Job is an instantiation of an Application. It contains Job Steps, which refer to Program attributes. *Job steps* contain Workunits.
- *Workunit*—A Workunit links the Program executable with input *Data*.
- *Data*—The *Data* objects contain information the Job uses as input. Multiple similar input Datas are grouped into Data Set.

### Logging on to the Grid MP Console

Ask your system administrator for the Grid MP Console URL.



**Management Console**

*User Authentication*

User Name

Password

Mode  ?

© 2000-2003 United Devices, Inc. All Rights Reserved.

Figure 1. Grid MP Console User Authentication Screen

At the User Authentication screen, you can choose between Simple and Advanced Mode. Simple Mode is suitable for all Job-related tasks. Advanced Mode displays additional information when the system encounters an error and displays GUIDs (ID numbers used to identify objects, such as Jobs, Devices, etc.). Unless you are troubleshooting or need a GUID, you should not need to log on in Advanced Mode.

## Submitting a Job to Run on Devices

To submit a Job via the Grid MP Console, you must:

- Have the Create Job privilege on the Application the Job will instantiate. It is also useful to have the Read Application and Read Program privileges so that you can view the Application or Program if necessary. The default Application Users User Group has all of the required privileges.
- In this example, you will create new Data Sets local to your Job and use them to create Workunits. To create local Data Sets, you need Update Job privileges, which you automatically have for any Jobs you create.

<p><b>NOTE</b> In the Grid MP platform, you can also create Global Data Sets, which can be used for other Jobs. To do that, you must have the Data Creator privilege.</p>
---

- Create the Job and Job Step(s)
- Create Data Sets containing the Data the Job will use
- Create Workunits from the Data Sets
- Ensure the Job is runnable

### ***Creating a Job and Job Step in the MP Console***

1. Log on to the MP Console
2. From the Grid MP home page, click **Job Creation Wizard**. The system displays the first screen of the wizard.



3. Select an Application from the list and click **Next** to continue.

	Application Name	Public Name	State
<input type="radio"/>	medium	-	Enabled, Open
<input type="radio"/>	example	-	Enabled, Open

total records returned: 2

**Figure 2. Job Creation Wizard, Application Selection Screen**

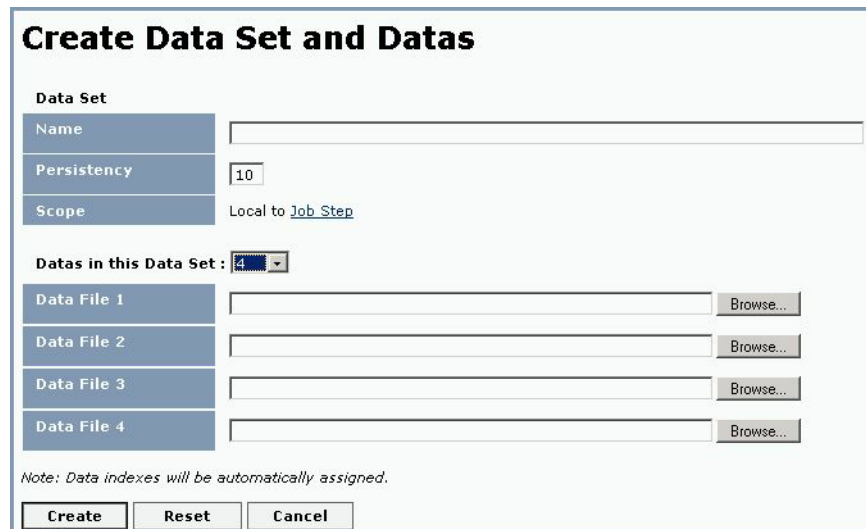
4. All fields on the second screen of the wizard contain defaults. If the defaults are acceptable, click **Create** to save the new Job. For help with field descriptions or other screen options, see the screen-level help by clicking **Help** from any screen.

Application	medium
Program	medium
Priority	10
Results per Workunit	<input type="checkbox"/> Infinite -- or -- 1
Errors per Workunit	1
Concurrent Dispatches per Workunit	1
Workunit CPU Timeout	1 00:00:00 (D) (HH) (MI) (SS)
Workunit Clock Timeout	1 00:00:00 (D) (HH) (MI) (SS)

**Figure 3. Job Creation Wizard, Details Screen**

5. The system displays a Request Completed screen, which means you've successfully created a Job and one Job Step. Click **Continue to Create Workunits**. From the Create Workunits from Data Sets screen you can create new local Data Sets or you can use existing Data Sets. This section details creating new local Data Sets. For help creating Workunits from existing Data Sets, use the screen-level help from the Data Set screens.

- Click **Wizard Create Data Set and Datas local to this Job Step**. The MP Console displays the Create Data Set and Datas screen.



**Create Data Set and Datas**

**Data Set**

Name

Persistency

Scope Local to [Job Step](#)

**Dats in this Data Set:**

Data File 1

Data File 2

Data File 3

Data File 4

*Note: Data indexes will be automatically assigned.*

**Figure 4. Create Data Set and Datas Screen**

- Choose the number of Dats you want to upload from the **Dats in this Data Set** drop-down list box. The screen will refresh with the corresponding number of Data fields.
- Provide a name for the Data Set and make changes to Persistency as needed. Persistency is a numerical representation of the likelihood that a Device will need to reuse the Data. If the persistency is 0, Devices will make no attempt to keep Data in this set locally on disk after using it. The higher the number, the greater the attempt to store the Data in cache. The number must be between 0 and 99.
- Click **Browse** for each Data field to locate and upload the Data files.

**NOTE** The Data Files you upload must have already been prepared for Grid MP usage with the buildpkg utility. For more information, see [“Creating Data Packages with buildpkg”](#) on page 34.

- Click **Create** to generate the Data and Data Set objects. The MP Console displays the Request Completed screen.

- Click **Continue to Create Workunits**. The MP Console displays the Create Workunits from Data Sets screen.

### Create Workunits From Data Sets

Add Data Sets to Job Step and automatically create Workunits.  
After pressing Create, Workunits will be generated from the cross product from all Datas in the the selected Data Sets.

**Actions**

- [Create Data Set local to this Job Step](#)
- [Wizard Create Data Set and Datas local to this Job Step](#)

**Job Step** Job Step

**Filter/Search Options**

<input checked="" type="checkbox"/>	Data Set Name	Persistency	URL first	Scope	Created
<input checked="" type="checkbox"/>	DataSet2	10	No	Local to Job Step	2004-01-24 01:15:58 UTC
<input checked="" type="checkbox"/>	dataset1	10	No	Local to Job Step	2004-01-24 01:14:38 UTC

*total records returned: 2*

**Figure 5. Create Workunits from Data Sets Screen**

- Select Data Sets from the list.
- Click **Create** to generate Workunits from the cross product of all Datas in the selected Data Sets. The MP Console displays the Request Completed screen. You have created a Job, Job Step, Datas, Datasets, and Workunits at this point.

## Checking Job Status

To determine if a Job is runnable, follow these steps:

- From the Grid MP home page, click **Manage Jobs**. The system displays a list of all Jobs you can manage.

### Jobs

List of all Jobs

**Actions**

- [Create Job](#)
- [Job Creation Wizard](#)
- [View Job Distribution by State](#)
- [Update Jobs](#)
- [Delete Jobs](#)

**Filter/Search Options**

Job ID	Description	Created	Application	Creator	State	Runnable	Workunits (done/total)	Results (succ/ unsucc)/ Errors	Priority
3	/agent_test...	2004-01-15 19:34:02 UTC	medium	test	Disabled	No	0/1 (0.00%)	0/0/1	1
4	/agent_test...	2004-01-15 19:34:45 UTC	medium	test	Disabled	No	0/1 (0.00%)	0/0/1	1
5	/agent_test...	2004-01-15 19:35:26 UTC	medium	test	Disabled	No	1/1 (100.00%)	1/0/0	1
101	job_for_init...	2004-01-15 21:38:04 UTC	example	test	Disabled	No	5/5 (100.00%)	8/0/0	1
102	job_for_init...	2004-01-15 21:40:51 UTC	example	test	Disabled	No	5/5 (100.00%)	10/0/0	1
103	job_for_init...	2004-01-16 00:05:20 UTC	example	test	Disabled	No	1/2 (50.00%)	6/0/5	1
104	job_for_init...	2004-01-16 00:22:15 UTC	example	test	Enabled	Yes	0/5 (0.00%)	0/0/1	1
105	Automaticall...	2004-01-24 00:56:56 UTC	medium	MPAdmin	Enabled	Yes	0/0 (0.00%)	0/0/0	10

*total records returned: 8*

**Figure 6. Jobs Screen**

2. Find the Job in the list on the Jobs screen and look under the **Runnable** column. "Yes" in this column indicates that the Job and Job Step are in the Enabled state, the Application and Program are in the Enabled, Closed, or Pending Delete status, and the current time is within the Job start and end times (if times were specified). If there is a "No" in the column, check the states of the Job, Job Step, Application, and Program, and check the Job start and end time.

**NOTE** Contact your system administrator if you think the Job is not running when it should. See "[How Are Jobs Scheduled on the Grid MP Platform?](#)," on page 3, for more information about how the Grid MP Dispatch Service schedules Jobs.

## Retrieving Results and Errors

To view a Job's results in the MP Console, you must meet the following requirements:

- Have a logon ID and password for the Grid MP platform
- Have Read Job privilege for the Jobs containing the results you want to retrieve. The default Application Users User Group has this privilege.

### Retrieving Results in the MP Console

1. Log on to the MP Console
2. From the Grid MP home page, click **Manage Jobs**. The system displays the Jobs screen. The Jobs screen contains a list of all the Jobs in the Grid MP platform. (see the "[Jobs Screen](#)" on page 25)
3. Click the number of results in the Results column corresponding to your Job. The system displays the Job Results screen.

**NOTE** You can also click the number of Errors in the Results column corresponding to your Job. This will take you to the Job Errors screen.

4. From the Job Results screen, you can view the time when the Workunits were sent and the time when results were received.
5. To download the results, click the link in the File Hash column.

## Retrieving Errors in the MP Console

1. Log on to the MP Console.
2. From the Grid MP home page, click **Errors**. The system displays the Errors screen. Only Errors resulting from Jobs for which you have the Read Job Privilege will display.

Details	Error Time	Source	Log File	Device Group	Device Name	Program Name	Platform	Agent Version
<a href="#">Unable to access data 07...</a>	2004-01-15 19:34:37 UTC	Agent	<a href="#">Download</a>	—	—	medium	i686-pc-linux-gnu	MP v4.1 - 3503
<a href="#">Unable to access data 07...</a>	2004-01-15 19:35:19 UTC	Agent	<a href="#">Download</a>	Default Device Group	guernsey	medium	i686-pc-linux-gnu	MP v4.1 - 3503
<a href="#">Unable to access data 2e...</a>	2004-01-16 00:07:38 UTC	Agent	<a href="#">Download</a>	Default Device Group	guernsey	example	i686-pc-linux-gnu	MP v4.1 - 3503
<a href="#">Unable to access data fs...</a>	2004-01-16 00:07:43 UTC	Agent	<a href="#">Download</a>	Default Device Group	guernsey	example	i686-pc-linux-gnu	MP v4.1 - 3503
<a href="#">Unable to access data a7...</a>	2004-01-16 00:07:48 UTC	Agent	<a href="#">Download</a>	Default Device Group	guernsey	example	i686-pc-linux-gnu	MP v4.1 - 3503
<a href="#">Unable to access data c3...</a>	2004-01-16 00:07:52 UTC	Agent	<a href="#">Download</a>	Default Device Group	guernsey	example	i686-pc-linux-gnu	MP v4.1 - 3503
<a href="#">Unable to access data a7...</a>	2004-01-16 00:07:57 UTC	Agent	<a href="#">Download</a>	Default Device Group	guernsey	example	i686-pc-linux-gnu	MP v4.1 - 3503
<a href="#">Unable to access data c3...</a>	2004-01-16 00:08:10 UTC	Agent	<a href="#">Download</a>	Default Device Group	guernsey	example	i686-pc-linux-gnu	MP v4.1 - 3503
<a href="#">Unable to access data fs...</a>	2004-01-16 00:23:21 UTC	Agent	<a href="#">Download</a>	Default Device Group	guernsey	example	i686-pc-linux-gnu	MP v4.1 - 3503

*total records returned: 9*

**Figure 7. Errors Screen**

3. A brief description of the Error displays in the Details column. Click the link to display additional information about the Error.
4. Click Download from the Errors screen or the Error details screen to save or download.

## Deleting Results and Jobs

To delete results in the MP Console, delete the Job or Job Step containing the results. To delete Jobs or Job Steps, you must meet the following requirements:

- Have a logon ID and password for the Grid MP platform.
- Have the Delete Job privilege on the Job you want to delete. The default Application Users User Group has this privilege.

### Deleting a Job in the MP Console

1. Log on to the MP Console.
2. From the Grid MP home page, click **Manage Jobs**. The system displays a list of all Jobs you can manage.
3. Locate the Job you want to delete in the list and click its corresponding index number. The system displays the Job details screen.

- Click **Delete** from the list of Actions. The system displays the Delete Job screen.



The screenshot shows a window titled "Delete Job" with a subtitle "Delete Job #5". It contains a table with the following data:

Description	./agent_test.pl
Job State	Disabled
Scheduled Execution Start	Not Specified
Scheduled Execution End	Not Specified
Priority	1

At the bottom of the window are two buttons: "Delete" and "Cancel".

**Figure 8. Delete Job Screen**

- Click **Delete**. The system prompts you for confirmation. Click **OK** to confirm and delete the Job. Click **Cancel** to return to the Delete Job screen without deleting.

## Appendix A: Application Administrator Tasks

The *Application User's Quick Guide* is intended for the Grid MP Application User—the user who uploads executables and data to the Grid MP platform, creates Jobs and submits them to run on the MP Agent machines, and monitors and downloads results. This section describes those tasks that must be performed to enable the Application User to work—tasks such as installing the command-line utilities and building Data files.

Throughout the previous chapters of this document, if a task had prerequisite steps that are usually performed by an Application Administrator, it was noted in the text and this Appendix was cross-referenced.

### mpbatch and ud\_mpirun Prerequisites

Two command-line submission utilities are included as part of the Grid MP platform. The mpbatch utilities allow users to submit Jobs from the command line. The ud\_mpirun utility allows users to submit MPICH Jobs in the Grid MP platform. This section describes the steps required to install and initialize the utilities before users can submit Jobs at the command-line.

<b>NOTE</b>	Installation files for mpbatch and ud_mpirun can be found on the Grid MP Installation CD-ROM. The instructions below provide commands and paths to files in the CD-ROM.
-------------	---

### All Submission Machines

All machines that will run mpbatch or ud\_mpirun utilities must have the following files. United Devices recommends copying these files to a shared network directory where MPI and batch Job submitters will be able to access them.

- buildpackage, buildmodule, batchmodule, and loader files—Utilities in the SDK ud\_mpirun and mpbatch will use to run your executable through the Grid MP platform. Copy these files to the machine you will use to run mpbatch or ud\_mpirun. The files are located in:
  - buildpackage, buildmodule, and loader files: /mnt/cdrom/sdk/UDsdk\_v4.1/tools/build

- batchmodule files: /mnt/cdrom/sdk/UDsdk\_v4.1/tools/mpbatch
- **uduserconf**—The configuration file that specifies necessary information, such as the location of the MGSI File Service and RPC Service. A sample configuration file is located in:

- /mnt/cdrom/sdk/UDsdk\_v4.1/uduserconf.sample

Before running ud\_mpirun or mpbatch utilities, copy the uduserconf file to the directory specified in your HOME environment variable and change the parameter values to match your configuration. On Linux, the file should be named “.uduserconf” (note that on Linux, the name begins with a dot). On Windows it should be named “uduserconf”.

The uduserconf contains the following parameters. The format of each parameter in uduserconf is a name and value pair, with # as the comment designator. The following line shows an example uduserconf parameter.

```
MGSI_USERNAME = myName
```

For assistance determining the correct parameter values, contact your system administrator.

**Table 5: uduserconf Parameters**

---

MGSI_FILESVR_URL	The URL to the MP File Service.
MGSI_XMLRPC_URL	The URL to the MGSI RPC Service.
MGSI_SOAP_URL	The URL to the MGSI SOAP Service
MGSI_USERNAME	The username for the MGSI RPC Service
MGSI_PASSWORD	The password for the MGSI RPC Service
MPBATCH_PATH	The relative path to the batchmodule
BUILDMODULE_PATH	The relative path to the buildmodule
BUILDPACKAGE_PATH	The relative path to the buildpackage
LOADER_PATH	The relative path to the loader utility
MPI_LISTEN_PORT	A port number or range of port numbers on the MPI root node to which the other nodes can communicate. Defaults to 12345-12360. The port that is actually selected is printed out by ud_mpirun when the Job is submitted.

---

## Initializing the mpbatch and ud\_mpirun Utilities

The batch submission system requires an Application named “batch” to exist in the Grid MP platform. MPI Jobs require an Application named “mpi.” These Applications



are created by the mpinit script, which must be run by an MP Admin User once before any users can submit Jobs via mpsub or ud\_mpirun.

**NOTE** When the Application objects are created by the mpinit script, it assigns Read and Update privileges to the objects for the Everyone User Group. If the Application and Program objects are deleted or if the access privileges associated with the objects are modified, it may become necessary to run the mpinit script again.

### **On Windows**

To run the script in the correct directory, perform the following steps:

1. Insert the Grid MP Installation CD-ROM.
2. Copy the batch submission files to your local directory.
3. Run the mpinit script.

**NOTE** You do not need to run the mpinit script since 'mpi' and 'batch' applications get installed during Grid MP installation. You will need to run the mpinit script only if the 'mpi' and 'batch' applications are deleted.

### **On Linux**

To run the script in the correct directory, enter the following commands:

1. Mount the Grid MP Installation CD-ROM:  
`mount /mnt/cdrom`
2. Copy the batch submission files to your local UD directory:  
`cp -R /mnt/cdrom/sdk/UDsdk_v4.1/tools/mpbatch/  
/anydirectory/mpbatch`  
Where *anydirectory* is where you want to copy the files.
3. Run the mpinit script.

**NOTE** You do not need to run the mpinit script since 'mpi' and 'batch' applications get installed during Grid MP installation. You will need to run the mpinit script only if the 'mpi' and 'batch' applications are deleted.

## UD\_MPIRUN

ud\_mpirun is a command-line submission utility that allows users to submit MPICH Jobs in the Grid MP platform.

Any machine can run ud\_mpirun. The machine must have the following files in addition to the prerequisites described in “[All Submission Machines](#)” on page 29:

- MPICH version 1.2.5 must be installed. The Windows NT and Linux versions of MPICH can be found in the following location:
  - /mnt/cdrom/sdk/3rd\_party/mpich/mpich-1.2.5-1a.tar.gz
- To successfully run MPI executables on the Grid MP platform on Linux, you must patch the MPICH library with the mpich-1.2.5-1a-patch-aa file, which is located in the UDsdk\_v4.1/tools/mpi directory. This requires compiling a custom MPICH Linux library and linking your original MPI executable against that library.

### ***Rebuilding the MPICH Linux Library***

On Linux, perform the following application development steps prior to running ud\_mpirun to rebuild your MPICH library to include the UD-specific patches to the ch\_p4 device type and to relink your application.

1. Mount the Grid MP CD-ROM.

```
mount /mnt/cdrom/
```

2. Navigate to a directory in which you want to untar the mpich tarball.

```
cd /tmp (or any directory you want to use for the temporary building process)
```

3. Untar the MPICH Linux tarball. The following command untars the tarball and creates the mpich-1.2.5 directory.

```
tar xzf /mnt/cdrom/3rd_party/mpich/mpich-1.2.5-1a.tar.gz
```

4. Change directory to the mpich-1.2.5 directory.

```
cd mpich-1.2.5
```

5. Patch the UD-specific files.

```
patch -p0 < /mnt/cdrom/UDsdk_v4.1/tools/mpi/mpich-1.2.5-1a-patch-aa
```

6. Configure MPICH to include the new UD-specific patch.

```
./configure --with-device=ch_p4 --prefix=/usr/local/UD/mpich -rsh=false
```

In the command above, you can set the prefix path to the directory of your choosing, however it cannot be the directory to which you untarred the mpich tarball.

7. Build and compile the MPICH library.

```
make
```

8. Copy the compiled files to the MPICH directory.

```
make install
```

The custom UD version of MPICH will now be installed into the directory /usr/local/UD/mpich.

### ***Relinking the MPICH Linux Library***

Once you have rebuilt the MPICH library, you must relink your application to use it. In most environments, one of the following commands can be used to recompile.

For C applications:

```
/usr/local/UD/mpich/bin/mpicc filename.c [options]
```

For C++ applications:

```
/usr/local/UD/mpich/bin/mpiCC filename.cpp [options]
```

## **Preparing the Grid MP Platform to Run Third-Party MPI Jobs**

Before users can run third-party MPI jobs, the following setup instructions must be completed. The example given below is for running LAM MPI Jobs on a Linux machine.

1. Ensure that all cluster nodes have access to a shared filesystem. In this example, the nodes are named “node1,” “node2,” etc. and the shared filesystem is mounted on /share.
2. Install the Grid MP Agent on the head node and use manual MP Agent installation to install the MP Agent on the shared filesystem. In this example, it is installed in /share/UD/mpagent.
3. In the Grid MP Console, create a new Device Group to contain the head node and make note of the Device Group name. In this example, the Device Group is named Head Node.
4. Move the head node of the cluster to the Device Group.

For each user who will be submitting work to the cluster, use the setcredential script to add the appropriate Credential objects for this cluster. For more information about using the setcredential script, see the *System Administrator's Guide*. 

```
$ perl setcredential create DeviceGroupName mpusername unix_username
```

You should see the following output from the setcredential script:

```
Looking up DeviceGroup 'Head Node'
```

```
Looking up User 'mpusername'
```

```
Creating Credential
```

5. Ensure that your shared directory and every containing directory has group and other permissions set to read and execute by using the `chmod` command. In this example, the shared directories are: `/share`, `/share/UD` and `/share/UD/mpagent`.

```
chmod go+rx /share /share/UD /share/UD/mpagent
```

6. Confirm that the permissions are correct:

```
ls -ld /share /share/UD /share/UD/mpagent
```

You should receive output like the following:

```
drwxr-xr-x  5 root    root      4096 Dec 17 12:05 /  
share/
```

```
drwxr-xr-x  3 root    root      4096 Dec 17 12:05 /  
share/UD/
```

```
drwxr-xr-x  2 root    root      4096 Dec 17 12:00 /  
share/UD/mpagent/
```

7. At this point, users should be able to submit Jobs using executables running third-party MPI instances, such as LAM-MPI. Instructions for Job submission can be found in “[Submitting Third-Party MPI Jobs with mpsub](#)” on page 16

## Creating Data Packages with buildpkg

A Data Package contains input files for a Program Module executable and descriptive information about those files that the MP Agent uses. Before a user can create a Data object in the Grid MP Console, a Data Package must be created, either manually, or by using the `buildpkg` tool. This section describes using the `buildpkg` tool.

The `buildpkg` tool automatically creates a Data Package and its contents, including the Data Package’s Package Manifest File (PMF). For more information about the PMF, see the *Application Developer’s Guide*.

The `buildpkg` tool is located in the `UDsdk_v4.1\tools\build` directory. To automatically create a Data Package and its contents using `buildpkg`:

1. Ensure that the `buildpkg` executable is in your path or in the current working directory.
2. Type the following at a Windows or Linux command line.

```
buildpkg myDataPackage.tar myInputData1 myInputData2
```

Where:

- `buildpkg` is the platform-specific `buildpkg` executable. Type one of the following:
  - `buildpkg.exe`—For Windows

- `buildpkg_i686-pc-linux-gnu`—For Linux
  - `buildpkg_powerpc-ibm-aix5.2.0.0`—For AIX
  - `buildpkg_sparc-sun-solaris2.8`—For Solaris
- `myDataPackage.tar` is the name of the Data Package you are creating.
  - `myInputData1` and `myInputData2` are the input files for the Program Module executable. There are no constraints on filename, file size, file format, or number of files.

In this instance, assuming that the files `myInputData1` and `myInputData2` exist, running `buildpkg` creates the file `myDataPackage.tar` that contains:

- `pmf.xml`
- `myInputData1.bz2`
- `myInputData2.bz2`

<b>NOTE</b> The <code>buildpkg</code> tool automatically compresses input files using bzip2 compression and the MP Agent automatically decompresses bzip2 input files.
--

Note that `buildpkg` has advanced features that are not discussed in this section. For a list of all `buildpkg` functions, including explanations and usage information and more information about the `buildpkg` tool, and for information about manually creating Data Package, see the *Application Developer's Guide*.

## Digital Signatures and MPBATCH and UD\_MPIRUN

You can employ digital signatures to authenticate the common types of Program Module executables sent to the MP Agents. There is limited compatibility, however, with `mpbatch` and `ud_mpirun`.

- **mpbatch**—You can run Jobs via the batch submission utility on a Grid MP platform utilizing the digital signature mode, but you cannot digitally sign the `mpbatch` executables themselves. To submit batch Jobs to digitally signed MP Agents, you must digitally sign the batch module executables before utilizing the `mpinit` script. For instructions about how to digitally sign files, see “Utilizing `udsign`” in Chapter 5 of the *Installation Guide*.
- **ud\_mpirun**—Running MPI Jobs is not compatible with use of the `udsign` technology. If your organization intends to run MPI Jobs on the Grid MP platform using `ud_mpirun`, you cannot digitally sign MP Agent modules and executables.



---

# Application User's Quick Guide Feedback Form

Please give us your feedback about the *Application User's Quick Guide*. Print this form, write your comments on it, and fax it to us at (512) 331-6235. Thank you for your feedback.

- 1. What operating system do you use?  
\_\_\_\_\_
  
  - 2. What is your job title?  
\_\_\_\_\_
  
  - 3. Please select the document for which you are providing feedback.  
 Application Developer's Guide     Installation Guide  
 Application User's Quick Guide     System Administrator's Guide
  
  - 4. Have you used this document to look for explanations about Grid MP platform concepts and technology?  
 Yes     No
  
  - 5. Have you used this document to perform a particular task?  
 Yes     No
- If yes, were the step-by-step procedures useful and accurate?  
 Yes     No

If the step-by-step procedures were not useful and accurate, what problem(s) did you have with them?

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

6. Did you use the Table of Contents and/or the Index in this document?

Table of Contents       Index

Were you able to find the information you were looking for?

Yes       No

If you were looking for information that you could not find, please describe.

---

---

---

---

---

---

---

---

---

---

7. Was the textual layout and design of this document useful and easy to navigate?

Yes       No

8. This document was delivered as an Adobe<sup>®</sup> .PDF file. Was this format useful?

Yes       No

Did you print this document and consult the hardcopy?

Yes       No

9. Do you have general or specific comments about this document or suggestions about ways in which this document can be improved?

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---





---