|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | |  | | --- | |  | |  | |  | | |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | | **Recovering the master Database or Master Device under ASE 12.5**  This TechNote describes the new procedure you need to use to load the master database, or recover from master database or device corruption in Adaptive Server Enterprise 12.5. The disaster recovery procedures differ from pre-12.5 servers. Contents 1. Background  2. Before You Begin  3. Loading an Older Copy of *master* Database  4. Recreating the *master* Database  5. Recreating the Master Device  6. Manually Setting the Backup Server Name Section 1. Background Starting with Adaptive Server version 12.5, there is no **buildmaster** program to build the master device. Rather, this functionality has been incorporated into the **dataserver** (unix) and **sqlsrvr** (Windows) programs. The server now allows you to create master devices and databases with 2K, 4K, 8K or 16K logical page sizes. Due to these and other changes, you cannot use the instructions provided in the *ASE Troubleshooting and Error Messages Guide* (EMTSG) for disaster recovery tasks like recovering the *master* database or device.  This TechNote explains how to perform disaster recovery in the 12.5.x server. It is applicable to ASE 12.5.0.1 IR and higher versions.  *Note*:  The EMTSG instructions still apply to the pre-12.5 servers.    Section 2. Before You Begin This document describes three key maintenance and disaster recovery tasks:   * Loading an older copy of your *master* database. This assumes that both the master device and *master* database are intact and free from corruption. * Recovering from a corrupted *master* database. This assumes that the master device is intact. * Recreating a master device and all its databases.   Some points to note before using this information:   * This material applies only to ASE 12.5.x and higher. It has been verified with ASE 12.5.0.1 Interim Release (IR), and it is recommended that you use the procedures with this release (or later). For 12.0 and prior versions, use the instructions provided in the *ASE Troubleshooting and Error Messages Guide*, "System Database Recovery." * This material assumes that your Adaptive Server was installed with your platform's default sort order. If you have installed a non-default sort order, you must ensure that your (restored) server uses the correct sort order and character set to reflect that in the dump. Refer to the *ASE Troubleshooting and Error Messages Guide*, "System Database Recovery," section titled "Valid Dump with Non-Default Sort Order" for details; and note that in Step 3 of that section, the **sybinit** utility has been replaced by the **dsedit/dscp** utilities. * All SQL command examples in this document use Transact-SQL syntax. All command-line examples are unix commands; Windows users can find the equivalent NT syntax in *ASE Utility Programs for Windows and Windows NT*. * Starting with 12.5, the **dataserver** command allows a space between option and parameter. * The examples in this document assume that                - the **dataserver** binary is located in *$SYBASE/bin/dataserver*               - the master device is *$SYBASE/d\_master.dat*           Replace this location and device name with those appropriate for your site.   * The server must be in single-user mode (that is, started with the **-m** flag) to load the *master* database. In this mode only the *master* database can be loaded. * You can only load a dump of *master* that matches your server level. Loading an older version dump to a newer server is not permitted. * You can use these procedures regardless of which version of the server you were using when you created your old master device. The server will find and correct any placement differences between the old and recreated databases. * After the load completes but before shutting down, the server does some post-processing to reconcile the newly loaded *sysdatabases* and *sysusages* tables against the information in the master device. At this time the server may print a variety of error messages regarding failures to use or find the *master* database, and/or attempts to insert duplicate keys or duplicate rows to *sysusages*. Ignore these messages; they occur only during the reconciliation phase, and will not affect the server's operation after it shuts down and is restarted.  Section 3. Loading an older copy of *master* database Use the following steps if your *master* database and the master device are intact, and you simply wish to load an older dump of your *master* database.  *Note*:  Be sure to read Section 2, Before You Begin.  Step 1: Put the Server in Single-user Mode  Shut down and restart the server with the **-m** flag, which places the server in single-user mode and sets up to load the *master* database:  **% $SYBASE/bin/dataserver -d $SYBASE/d\_master.dat -m**  Step 2: Establish the Backup Server Name  This step is necessary to ensure that Adaptive Server has access to its backup server for dumps and loads. Follow the instructions detailed in Section 6, Manually Setting the Backup Server Name.  Step 3. Load the master Database  Issue the following **isql** command:  **1> load database master from "master\_db\_dump"**  **2> go**  Adaptive Server shuts itself down after the load is complete.    Section 4. Recreating the master Database Use this procedure when the current master device is usable, but you are unable to use the server because of *master* database corruption. These steps enable you to create a new *master* database and reload it from backup.  Step 1. Create a New master Database  The approach to creating the new *master* database depends on the extent and nature of the corruption. Three different scenarios are possible:   * Basic recreation, which is sufficient if only the data in *master* was affected. The server reads the master device to determine page and device sizes. * Recreating when the device's configuration area is corrupted. You will need to provide page and device sizing information. * Recreating when the master database allocation pages are also corrupted. All corrupt or unallocated extents on the device are allocated to *master*.   *Basic Recreation of master Database*  This command instructs the server to read the device's configuration area to obtain page size and device size and determine where to place the *master* database:  **% $SYBASE/bin/dataserver -d $SYBASE/d\_master.dat -w master**  The server creates a *master* of the same size, and in the same locations on disk, as the database it is replacing. It will NOT have the old database's data!  Instead, it contains a default set of data that you will replace later via **load database**. The default data includes information about any databases existing on the master device (but no other devices).  It also has minimal system information, including a login for *sa* with a *null* password.  This process produces a large number of "upgrade" messages tracking the progress of database creation which are helpful in troubleshooting any problems. They are "upgrade" messages because the server creates a new *master* database by "upgrading" the device.  *Note*:  If the configuration area is corrupt or unavailable, this command returns the message: "The configuration area in device 'xxx' appears to be corrupt. The server needs this data to boot, and so cannot continue." If this occurs, continue with the instructions below.  *Recreation with a corrupt configuration area*  The "Basic Recreation" process above may fail if the device's configuration area has become corrupt.  If so, you must supply sizing information. You will need two parameters: the page size (you need to know what this was), and the device size, which you can determine directly from the device:  **% ls -l $SYBASE/d\_master.dat**  Divide the size shown by the page size (2048, say) to obtain the number of server pages, by 1024 to obtain KB, or by 1048576 to obtain MB.  Provide this information on the command line as follows:  **% $SYBASE/bin/dataserver -d $SYBASE/d\_master.dat -w master**  **-z page\_size  -b device\_size**  For example, if your page size is 2K and the device size is 51204 server pages (100 MB, plus 8K space for the configuration area), the command looks like this:  **% $SYBASE/bin/dataserver -d $SYBASE/d\_master.dat -w master -z 2k -b 51204**  You may also specify the device size as Kb, Mb, or Gb; for example, "-b 100M".  *Recreation when master database allocation pages are corrupted*  If the above procedures for recreating the *master* database fail, the database's allocation pages are corrupt. (This may happen, for instance, if the database device was inadvertently written over by a completely different file.)  In this case, you can force the server to allocate all corrupted or unallocated extents to the *master* database:  **% $SYBASE/bin/dataserver -d $SYBASE/d\_master.dat -w master -f**  This allocates ALL corrupted or otherwise unrecognizable extents to the *master* database.  Depending on the extent of your master device corruption, and how much free space it originally had, this will probably leave *master* much larger than it needs to be, causing it to occupy space that used to belong to other databases like *model, tempdb,* and *sybsystemdb*. We will consider recovering from that situation later.  *Note*:  You may combine the **-f, -b,** and **-z** options as necessary.  Step 2. Restart the Server in Single-user Mode  The server shuts down after recreating the *master* database. Restart it with the **-m** flag, which places the server in single-user mode and sets up to load the *master* database:  **% $SYBASE/bin/dataserver -d $SYBASE/d\_master.dat -m**  Step 3: Account for Missing Databases (if you used the **-f** option)  *Note*:  You only need this step if you used the **-f** option in Step 1 to recreate the *master* database due to allocation page corruption. If you did not use **-f**, proceed to Step 4.  Recall that the **-f** command line option could make the new *master* larger than needed at the expense of other required databases on the master device. You will need to check for these databases before proceeding. This step has many possible permutations, so you must know what databases *should* be on the master device to perform this step. For example, if you had moved *tempdb* to a different device, you will not need *tempdb* on the master device.  If upgrading, you may well have created *sybsystemdb* on a device other than *master*; if so, you will not need to account for *sybsystemdb*.  Log in as *sa*, and check the databases currently on the master device:  **1> select name from sysdatabases**  **2> go**  Do you see all the databases that should be on the master device?  If so, skip the rest of this step. Otherwise, you will need to determine which databases are missing *and* how big they should be, then obtain the free space needed to recreate these databases.  The following **isql** script obtains the required space by removing it from the end of the *master* database. In order, it   * establishes how many logical pages the missing databases need * subtracts that number from the pages that *master* occupies * removes disk usage entries for parts of *master* above that limit * restricts the highest logical chunk of *master* such that its total size leaves the required number of pages free.   You will need to provide the required space value, denoted as "*@needed\_mb*".  *Note*:  This sample script is provided to assist you with the disaster recovery task. It is not officially supported by Sybase.          1> declare @needed\_mb int, @needed\_pages int, @master\_end int,          2>      @pgspermb int          3> select @pgspermb = (1048576 / @@maxpagesize)          4> select @needed\_mb = 12 -- **replace '12' with required space value**          5> select @needed\_pages = @needed\_mb \* @pgspermb          6> select @master\_end = sum(size) - @needed\_pages          7> from master.dbo.sysusages          8> where dbid = 1          9> if (@master\_end > (6 \* @pgspermb))          10> begin          11>     delete master.dbo.sysusages          12>     where lstart > @master\_end          13>     update master.dbo.sysusages          14>     set size = @master\_end - lstart          15>     where dbid = 1          16>     and lstart = (select max(lstart) from master.dbo.sysusages          17>                  where dbid = 1)          18> end          19> else          20> begin          21>     print "Can't take enough space from the master database!"          22>     print "Need to find %1! pages", @needed\_pages          23>     print "That would leave master with %1! pages", @master\_end          24>     print "Cannot continue."          25> end          26> go  *Note*:  If the procedure fails, your master device is not big enough to hold all the databases you are trying to create.  Check the required MBs of space that you specified. If it is correct, it may be necessary to create a new master device using the instructions in Section 5, Recreating the Master Device.  You now have enough space to recreate your required databases. Create them one at a time. For example:  **1> create database model on default = 3**  **2> go**  Repeat for each database. Then shut down the server, and restart it in single-user mode using step 2 above.  Step 4. Establish the Backup Server Name  This step is necessary to ensure that Adaptive Server has access to its backup server for dumps and loads. Follow the instructions detailed in Section 6, Manually Setting the Backup Server Name.  Step 5. Load the master Database  Issue the following **isql** command:  **1> load database master from "master\_db\_dump"**  **2> go**  Unlike during a normal database load, the server may need to perform a lot of extra work, because *master* contains information about the identity and location of your other databases; that information may have changed for this master device, and the server must check and update it as necessary.  At the end of this step, the server contains correct disk usage information about the master device. This may differ from the information in *sysusages* in the dump, so the server will find and correct both the size and location information for the databases.  Any entries for parts of databases that don't actually exist on master will be removed.  During post-processing from the load, you may see one or more errors from the server. Read Section 1, Before You Begin, for more information about these errors.  After checks and validations are complete, the server will shut down. You may now restart it normally.  Step 6. Did you recreate any databases in step 3 above?  If you recreated databases using the procedure in step 3 above, load those databases.  You must restart the server *without* the **-m** flag in order to accomplish this. Section 5. Recreating the Master Device Use these steps when the disk that used to contain your master device is not accessible, and you need to start over with a new device.  This situation is somewhat similar to the scenario in Section 4 above in which the master device becomes so corrupt that you need to use the **-f** option, because you will need to know what databases used to be on your master device and how big they were so you can validate and recreate them as necessary.  Step 1. Create your new master device  When creating the new master device, make sure you use the same page size as your old master device and make the new device at least as large as the old one. The following example creates a device with a 2048-byte logical page size, and total size 100 Megabytes + 8 Kilobytes (the 8 KB is extra space for the configuration area.)  **% $SYBASE/bin/dataserver -d $SYBASE/d\_master.dat -z 2k -b 51204**  Use the **-s** option with this command to specify the server name. You can also specify the "-b" size in Kb, Mb, or Gb. In the above example you would use "-b 100.00782M". Without one of the K, M, or G modifiers, the default device size is expressed in server virtual pages, 2048 bytes each.  At device creation, the server issues large numbers of "upgrade" messages tracking its progress; these messages help troubleshoot any problems. They are upgrade messages because the server creates a new installation by doing an "upgrade" of a device that it has just created.  When finished, the server shuts down. You now have a *master* database containing minimal system information, including an *sa* login whose password is *null*, and minimally sized *master, model, tempdb*, and *sybsystemdb* databases.  Step 2. Put the Server in Single-user Mode  Shut down and restart the server with the **-m** flag, which places the server in single-user mode and sets up to load the *master* database:  **% $SYBASE/bin/dataserver -d $SYBASE/d\_master.dat -m**  Step 3. Establish the Backup Server Name  This step is necessary to ensure that Adaptive Server has access to its backup server for dumps and loads. The new master database contains a default Backup Server entry of *SYB\_BACKUP* for *srvnetname*, which is probably wrong. Since the  **sp\_addserver** procedure is not available at this time, log in to the server as *sa* and update *sysservers* directly:  **1> update master.dbo.sysservers**  **2> set srvnetname = "backup\_server\_name"**  **3> where srvname = "SYB\_BACKUP"**  **4> go**  (1 row affected)  Step 4. Load the *master* Database  Issue the following **isql** command:  **1> load database master from "master\_db\_dump"**  **2> go**  The server inspects the master device and makes any corrections needed in the newly loaded *sysdatabases* and *sysusages*. These corrections affect only the master device, since that is the only device that changed -- the server assumes that all your other devices are undamaged and need not be inspected.  After this step, it is possible that your new master device contains database entries for databases that also exist on other devices in your system.  This may happen if you moved *tempdb* to a different device, or created *sybsystemdb* on a different device.  The server recognizes and handles this situation: if it finds pre-existing entries for those databases on other devices, it presumes that the existing entries are correct and does not change them.  During post-processing from the load, you may see one or more errors from the server.  Please read Section 2, Before You Begin, for more information about these errors.  After the checks and validations are complete the server shuts down. You may now restart it normally.  Step 5. Check that the Databases on Master Device Are Correct  When you created a new master device in step 1 above, the server created only its default set of databases, with minimal data. You will almost certainly need to load dumps of the databases (notably *model*) that used to be there.  Are the databases on your new master device large enough to hold the dumps you will be loading into them?  Are all the necessary databases present? Is there any obsolete data that you need to clean up?  Log in as **sa** and inspect the databases on your system:  **1> declare @pgspermb int**  **2> select @pgspermb = 1048576 / @@maxpagesize**  **3> select "db name"=db\_name(dbid), dbid, "size"=sum(size) / @pgspermb**  **4> from master.dbo.sysusages**  **5> group by dbid**  **6> go**  This command shows you all the databases present on your system, and their total size. Note that the *size* column in the output is expressed in Megabytes.  Does this list contain any entries where database name is *null*? These *sysusages* entries don't have any matching entries in *sysdatabases*; they are unnecessary and should be deleted.  (You may be specially susceptible to this if you upgraded from pre-12.0 versions, and created *sybsystemdb* on the older version; *sybsystemdb* will have a different dbid than the default *dbid*.)  To remove these entries, use a script like the following:  **1> exec sp\_configure "allow updates", 1**  **2> go**  **1> delete sysusages**  **2> where db\_name(dbid) is null**  **3> go**  **1> exec sp\_configure "allow updates", 0**  **2> go**  Are any databases missing?  Create those databases. Are the databases large enough?  If not, alter them to be at least large enough to hold the dumps. (It is okay if they are too large; the server simply clears the excess space.)    Section 6. Manually Setting the Backup Server Name This procedure updates the *sysservers* table and is needed to ensure that Adaptive Server can access the correct backup server to carry out dumps and loads. Use it with the instructions for Sections 3 and 4.  Execute the following **isql** commands in Adaptive Server:  **1> use master**  **2> go**  **1> select srvname, srvnetname from sysservers**  **2> where srvname = "SYB\_BACKUP"**  **3> go**  There are three possible outcomes to this query. Take the appropriate action below depending on the outcome:      |  |  | | --- | --- | | Outcome | Action | | ASE returns a single row and   *srvnetname* contains the correct   Backup Server name | No action is needed. | | ASE returns a single row  but the *srvnetname* is not the  correct Backup Server name | Issue the following **isql** command:  **1> update sysservers**  **2> set srvnetname = "backup\_server\_name"**  **3> where srvname = "SYB\_BACKUP"**  **4> go**  where *backup\_server\_name* is the name of the Backup Server   as it appears in the *interfaces* file*.* | | ASE returns 0 rows | Issue the following **isql** command:  **1> sp\_addserver SYB\_BACKUP, null,**  **2> backup\_server\_name**  **3> go** | | | |
|  | [Back to Top](http://webnt/technotes/docs/tech/ase/Recovering%20the%20master%20Database%20or%20Master%20Device%20under%20ASE%2012_5.htm#top)   |  |  | | --- | --- | |  | **Document Attributes** |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | |  |  | | --- | --- | | **Content Id:** 1016939 | **Last Revised:** April 05, 2002 | |  |  | | **Authorization:** Public | **Focus:** Technical | | **Career:** DBA | **Career Interest:** Developer |  | | **Document Library:** Technote | **Products:** Adaptive Server Enterprise |  | | **Solutions:** Not Solution Specific | **Expertise Level:** 4-all |  | | **Technical Topics:** Database Admin |  |  | |

|  |
| --- |
|  |
| |  | | --- | | [*© Copyright 2002 - Sybase Inc.*](http://www.sybase.com/detail?id=1011207)  [**Help**](http://webnt/help_menu?id=19441) **|** [**Feedback**](http://webnt/feedback) **|** [**Contact Us**](http://webnt/contact_us) **|** [**Legal**](http://webnt/detail?id=1003650) **|** [**Privacy**](http://webnt/detail?id=1003649) | |
|  |