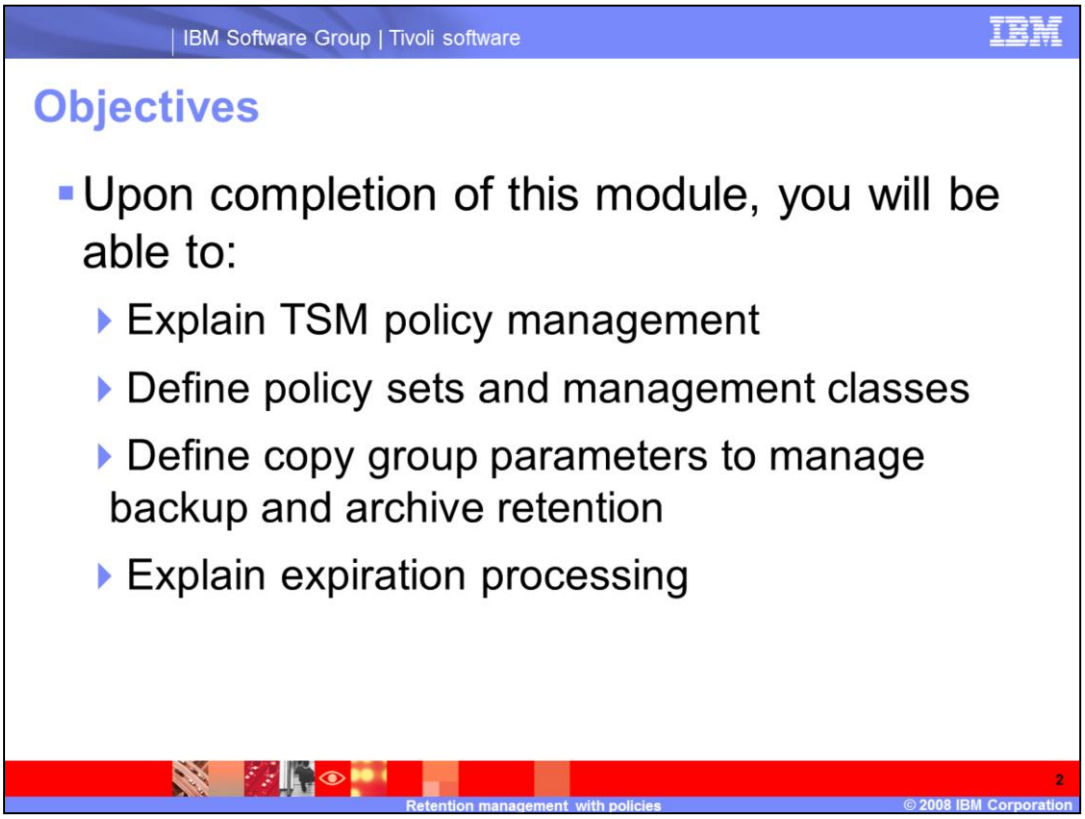
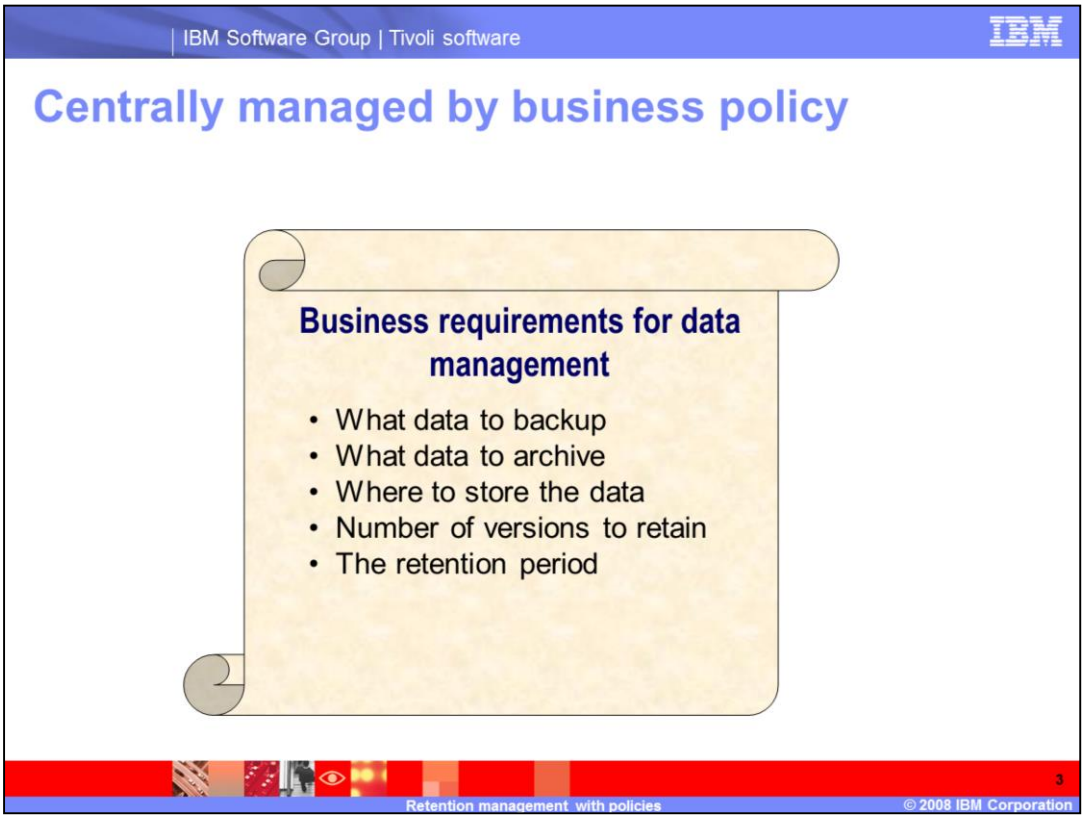
Slide 1 **Retention management with policies**

Welcome to the IBM Education Assistant training for IBM Tivoli Storage Manager version 5.5. This module covers how to manage backup-archive retention values with Tivoli Storage Manager policies. In this training, Tivoli Storage Manager is also referred to as TSM.



Slide 2 **Objectives**

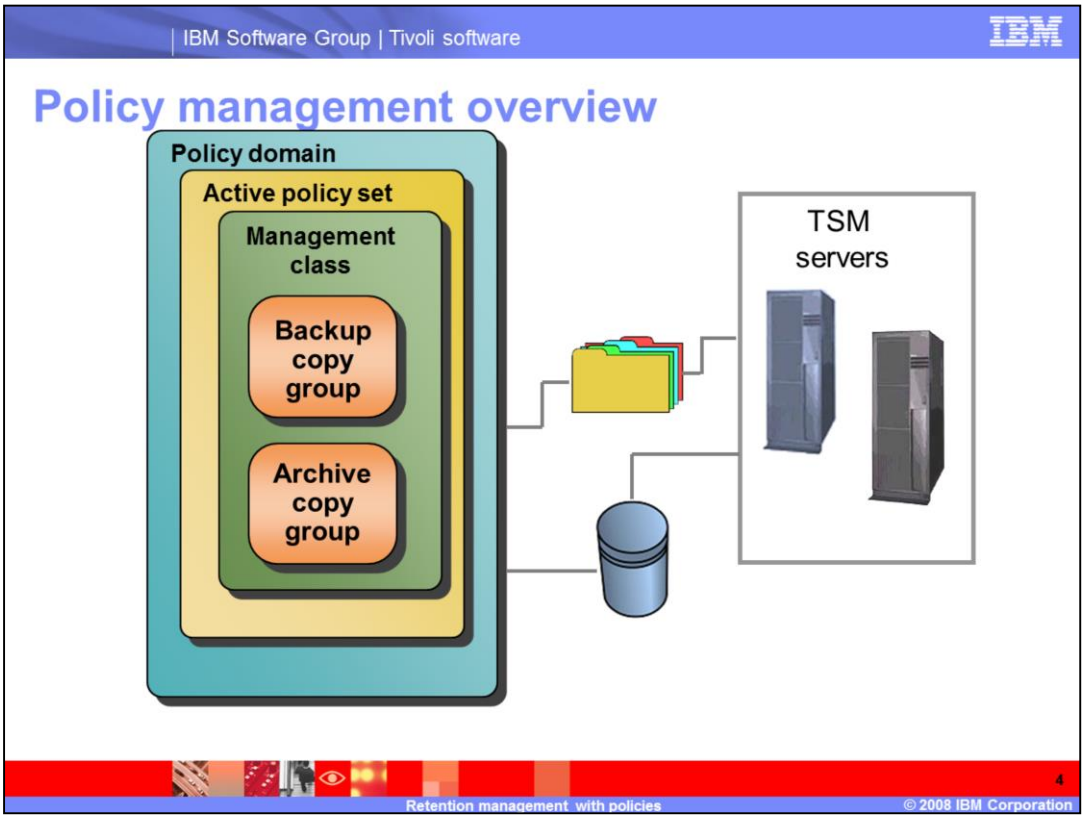
Upon completion of this module, you will be able to explain TSM policy management. Define policy sets and management classes. Define copy group parameters to manage backup and archive retention. And explain expiration processing.



Slide 3 **Centrally managed by business policy**

Questions such as what data to backup or archive, where to store the data, for how long, the number of versions to retain will determine your backup-archive policies.

So whether it's business requirements that are preferences or mandatory regulations, the customized policies can be defined for each business requirement.



Slide 4 **Policy management overview**

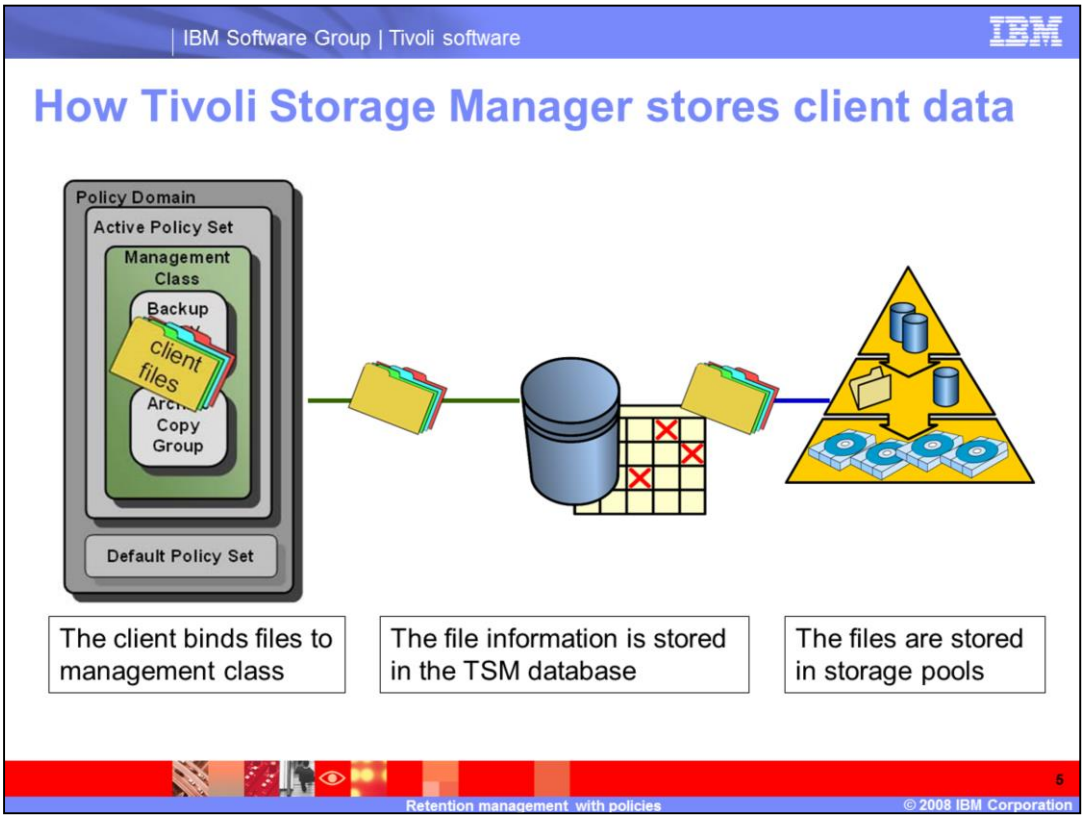
Think of policy management as stacking boxes inside of boxes. For example, in this graphic you can see that the largest container, the Policy Domain, contains the Active Policy Set which contains the Management Class, which contain Backup Copy Groups and Archive copy Groups.

So, policies are created by the administrator and they are stored in the database on the server. They can be updated and retroactively applied to already managed data. You can have multiple policies depending on business needs.

So, let's start with the largest box, the policy container for all policy components. The Policy Domain is a set of rules applied to a group of nodes managed by the same set of policy constraints. This is defined by the policy sets. A node must only be defined to one policy domain per server. However, a node can be defined to more than one Tivoli Storage Manager server.

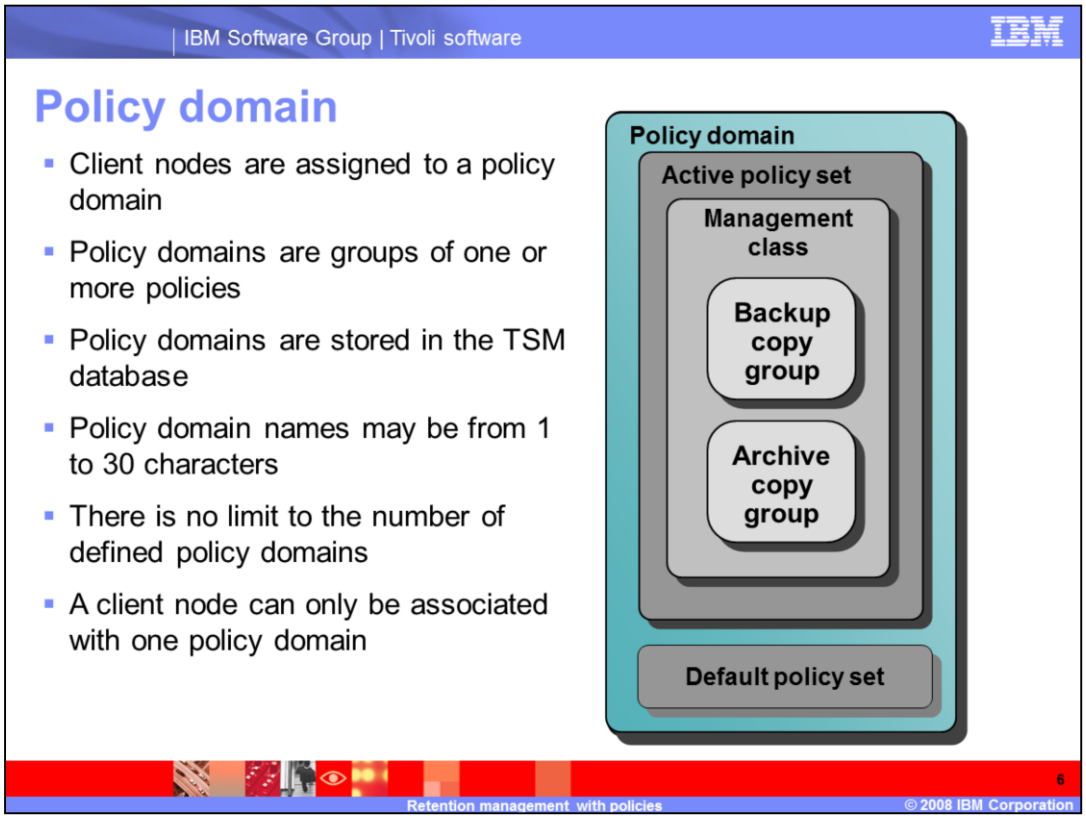
A Policy Set is a collection of management class definitions. A policy domain can contain several policy sets, however, only one policy set in a domain can be active at any given time. And that is the key there, is your active policy set.

The Management Class, or MC as it is known, is a collection of management attributes describing backup-archive characteristics. There are two sets of Management Class attributes, one for backup and one for archive. And these set of attributes are called a copy group. So you have a backup copy group and an archive copy group.



Slide 5 **How Tivoli Storage Manager stores client data**

So once the retention requirements are determined, you will then associate data to be backed up or archived with the policy. The client binds files to management class, the file information is stored in the TSM database, and then the files are stored in storage pools.



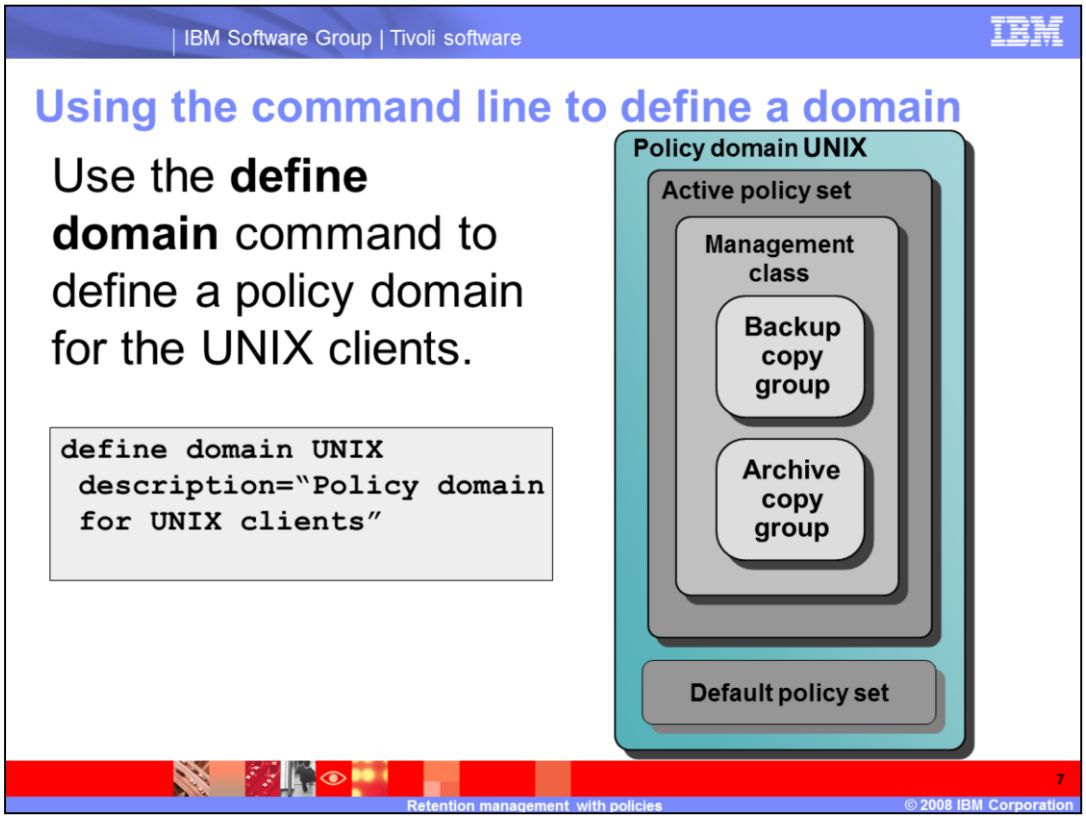
Slide 6 **Policy domain**

A policy domain provides you with a logical way of managing backup and archive policies for a group of nodes with common needs. These common needs are the requirements you have determined for your business. The Policy Domain is a collection of one or more nodes or clients and one or more policies. The domain is an object stored in the TSM database with a name from 1 to 30 characters. And the names should be meaningful.

There is no limit to the number of policy domains that can be defined on a TSM server.

A client node can be associated with only one policy domain on any specific TSM server. However, a client or node can be registered to more than one TSM server. Each domain then can have one or more clients or nodes associated with it. The clients or nodes can be running on the same or different platforms.

A policy domain also contains a grace period backup and an archive retention period. This grace period acts as a safety net to insure that data backed up or archived in a storage pool is not inadvertently deleted.



Slide 7 **Using the command line to define a domain**

The example shown here is the define domain command and is being used to define a policy for the UNIX clients. So the command you have here is define domain UNIX description="Policy domain for UNIX clients", with the description in quotations.

Let's talk about the different parameters that can be used with the command line.

You have the domain name, which specifies the name of the policy domain to be defined. This

parameter is required. The maximum length of this name is 30 characters.

You have the description. This specifies a text string that describes the policy domain. The parameter is optional, but it is suggested to provide a meaningful description. The maximum length of the description is 255 characters.

Then you have backup retention. This specifies the number of days (from the date the

backup versions became inactive) to retain backup versions of a file that no longer

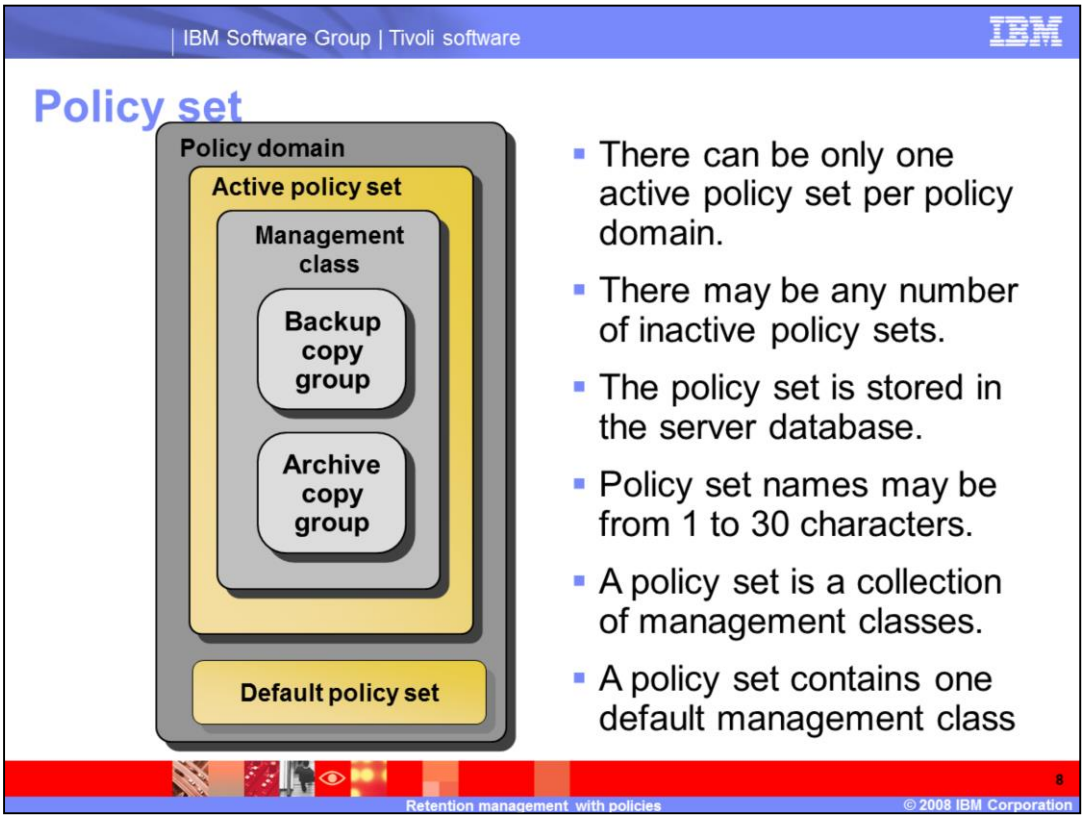
exists on a client system. This is your grace period for backups.

You have archive retention, which is the grace period for archives. It specifies the number of days (from the date of archive) to retain archive copies that are bound to a management class that no longer exists on the client system.

Related commands include: update domain and delete domain.

Remember, the policy domain also contains a grace period backup and an archive retention period.

You can also use the Administration Center to define policy domains.



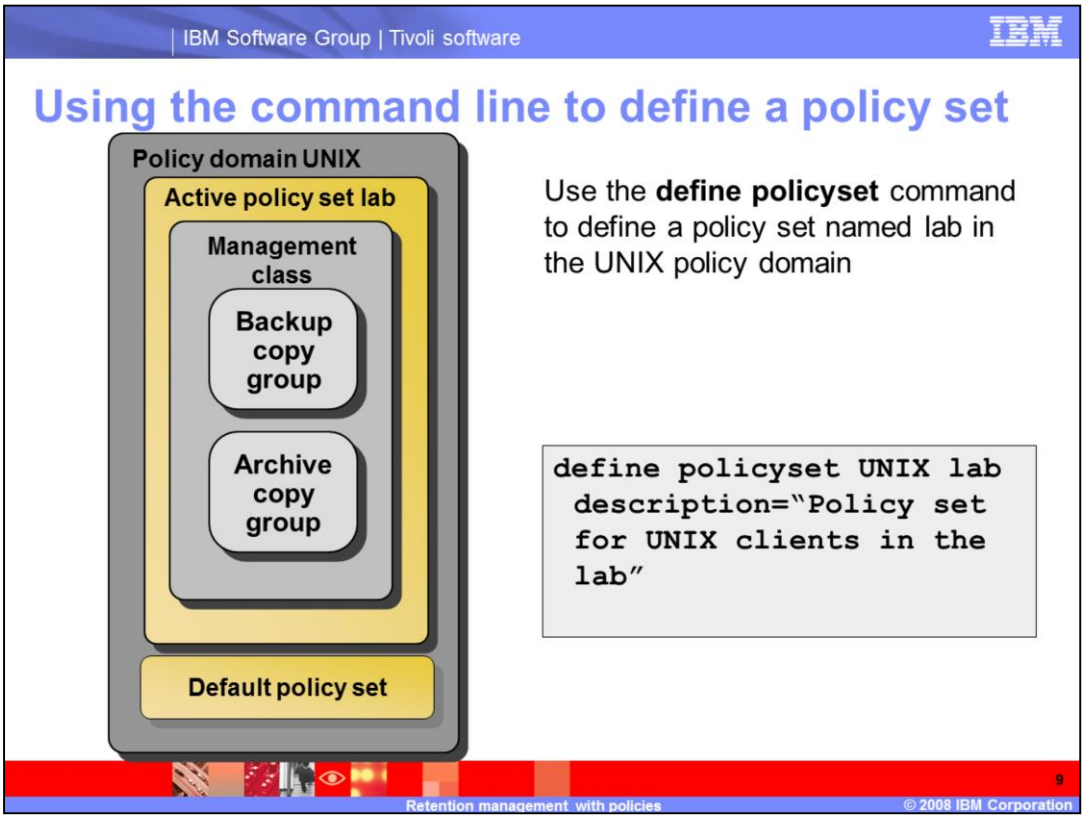
Slide 8 **Policy sets**

Remember, there can only be one active policy set per policy domain. You can have several inactive policy sets, but only one policy set will be active at any given time.

Each policy set contains a default management class and can contain any number of

additional management classes. Policy sets are used to implement different policies based on user and business requirements.

You must assign a default management class for a policy set before you can activate that policy set. It is a good idea to have a default management class with both an archive copy group and a backup copy group.



Slide 9 **Using the command line to define a policy set**

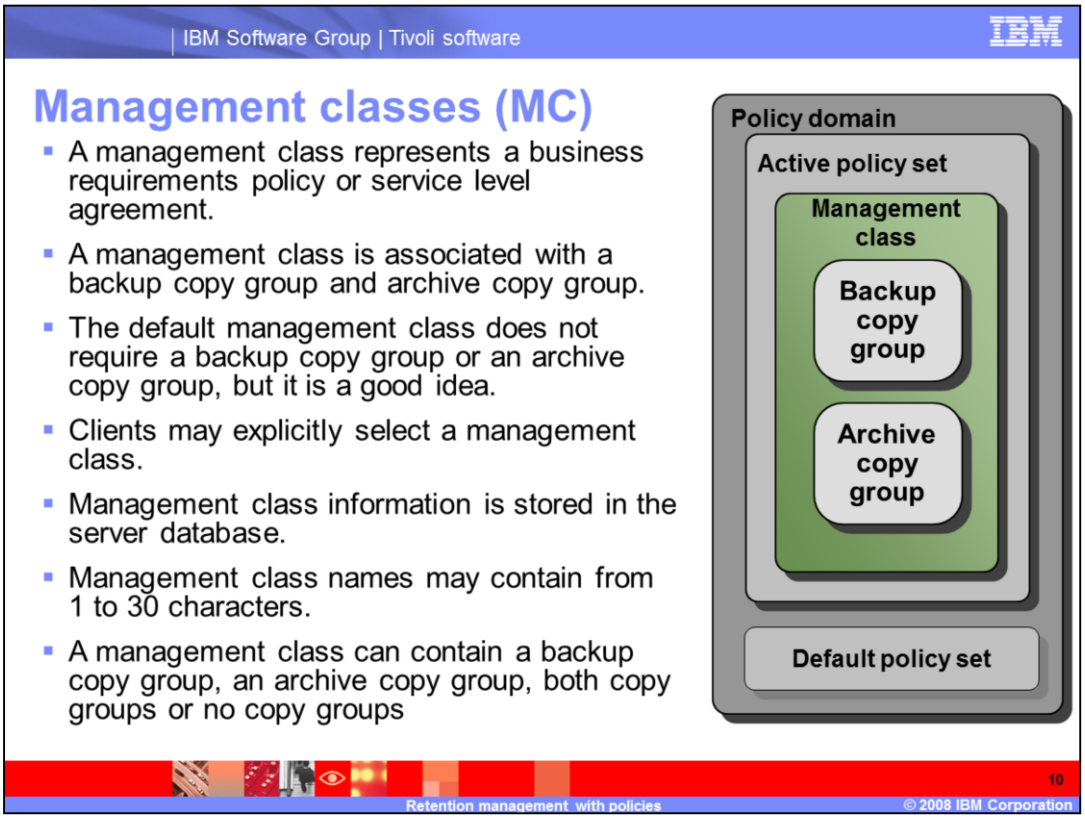
This example uses the policy domain of UNIX, and you are going to create a policy set for the lab.

So the command used is **define policy set UNIX lab description="Policy set for the UNIX clients in the lab"**, with the description in quotation marks.

The parameters you use are domain name, which is a required parameter, and this specifies the name of the policy domain to which the policy set belongs. In this case, UNIX. Set name. This specifies the name you want to assign to the policy set. This parameter

is also required and has a maximum length of 30 characters.

And again, your description field. It will describe the new policy set using a text string. This parameter is optional, but very helpful.



Slide 10 **Management classes or MCs**

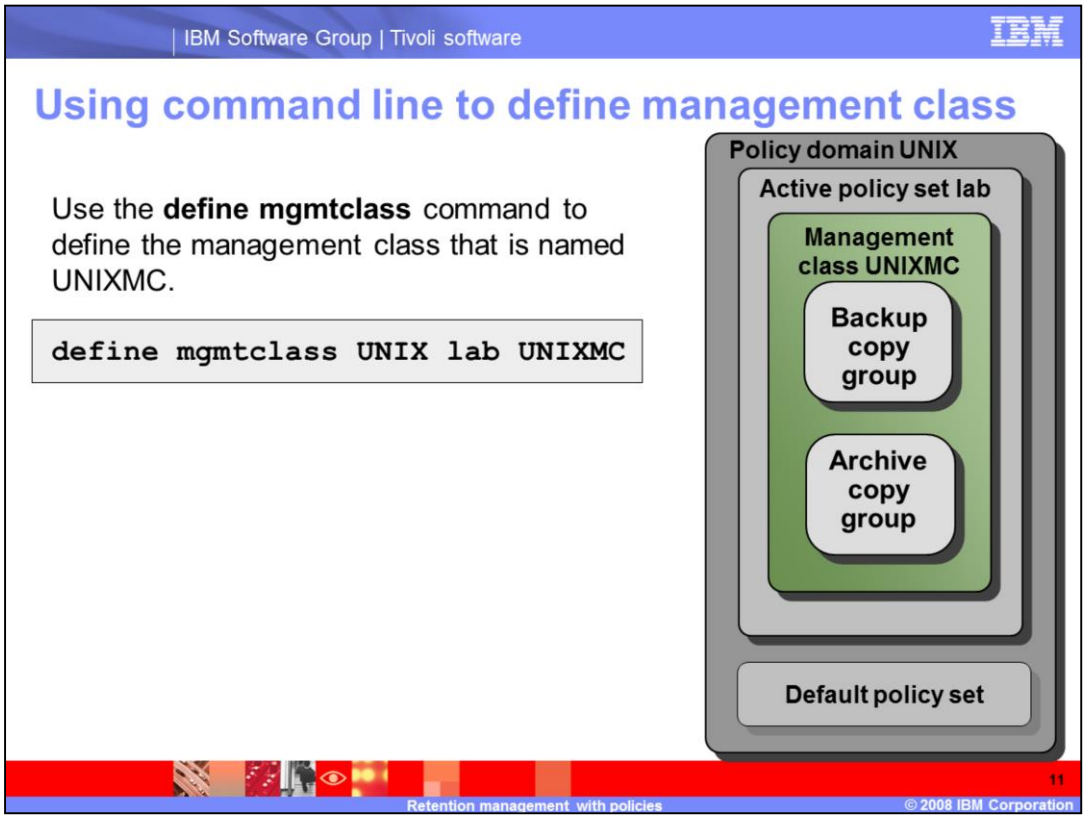
Remember, the management class is the representation of a business requirement or service level agreement.

The management class associates backup and archive groups with files and then it specifies if and how client node files are migrated to the storage pools. You can bind (which means,

associate) files to a management class using the include-exclude list. This would include and exclude files and directories from being backed up.

The default management class does not require a backup copy group or an archive copy group, but it is a really good idea to have one.

So you can either have a backup copy group, an archive copy group, or both copy groups, or no copy groups.

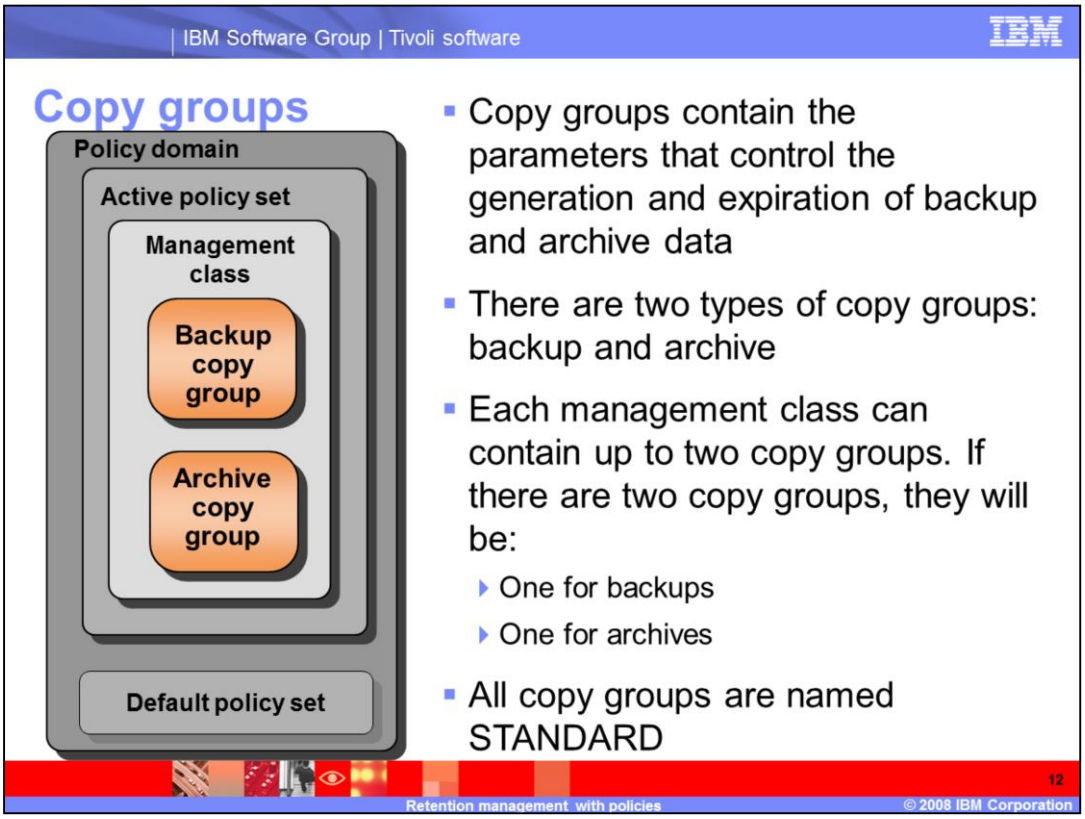


Slide 11 **Using command line to define management class**

Again you are using the policy domain of UNIX, the active policy set is lab, and you are going to create a management class called UNIXMC.

The command is **define mgmtclass UNIX lab UNIXMC description=”MC for UNIX lab clients”**. The description is in quotation marks**.**

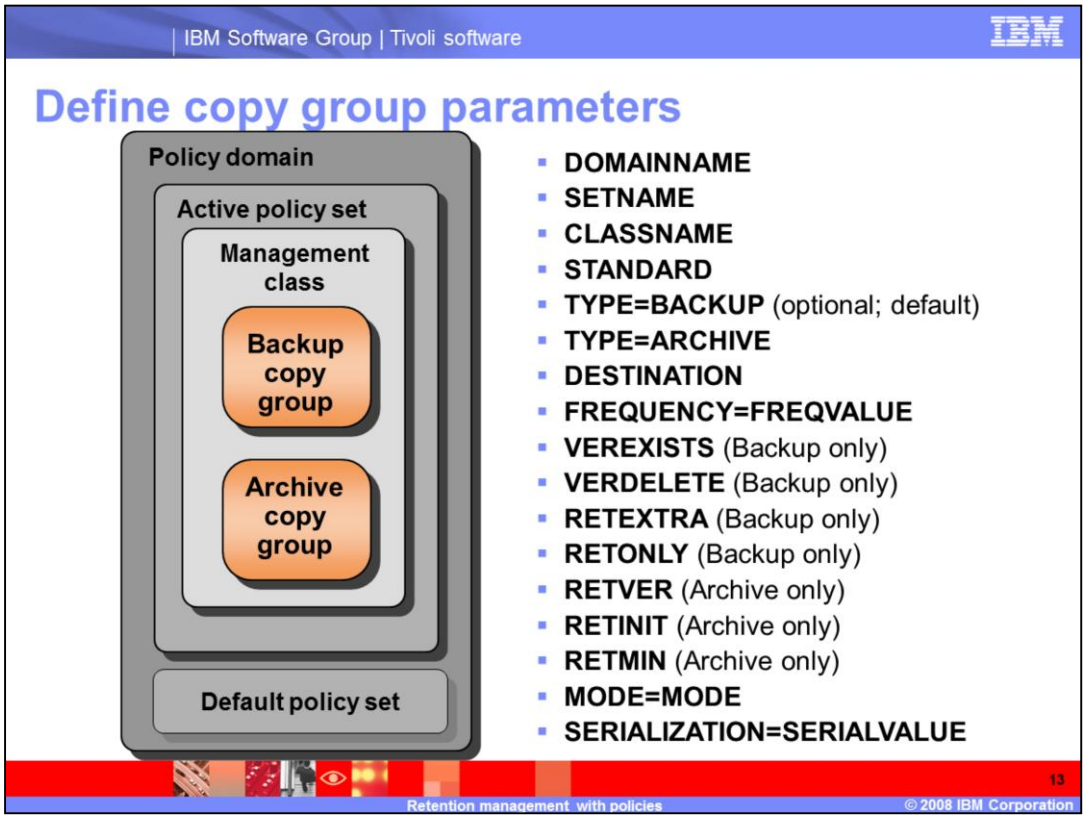
The parameters you use are the domain name and the required parameter. Set name, another required parameter. Class name, which will specify the name of the new management class. It is also required. And then the description field.



Slide 12 **Copy groups**

The copy groups are the containers for the specific storage management attributes. This tells the server how to manage the backed up or archived files.

Now you can have up to two copy groups; one for backups, one for archives. And all copy groups are named standard.



Slide 13 **Define copy group parameters**

If you have been wondering where the detailed parameters are for your policy, they are here with the define copygroup command.

So you are going to use the define copygroup command to define a new backup or archive copy group. Now these are held within the management class, which is in the active policy set, which is in the domain.

So you will have your domain name, the set name, and the class name as the required parameters.

Next you have STANDARD. Standard specifies the name of the copy group. Remember copy groups must be named STANDARD. Since this is the default value, it is therefore an optional parameter.

Type=Backup. This is the default. If you are defining a copy group, the default is for a backup copy group, therefore it is optional. However, if you are defining an archive group, you must include Type=Archive. Specifies that you want to define an archive copy group. The default parameter is ARCHIVE

The destination will specify the name of the primary storage pool.

FREQuency=freqvalue, this specifies the minimum interval, in days, between successive backups. And this is an optional parameter.

The next couple parameters are only used when defining a backup copy group.

The first one is VERExists, which is for versions exist, and it specifies the maximum number of backup versions to retain for files that are currently on the client file system. So number of versions you would retain, and the default here is 2.

VERDelete, or versions delete, specifies the maximum number of backup versions to retain for files that have been deleted from the client file system after being backed up with TSM. The default value here is 1.

RETExtra, would be retain extra, specifies the number of days to retain a backup version after the version becomes inactive. A version of a file becomes inactive when the client stores a more recent backup version, or when the client deletes the file from the workstation and then runs a full backup. The server deletes inactive versions based on retention time even if the number of inactive versions does not exceed the number allowed by the VERExists or VERDelete parameters. The default value here is 30 days. In a few slides you will see the expiration processing for inactive files.

And finally the RETOnly, the last of the backup only parameters. This specifies the number of days to retain the last backup version of a file that has been deleted from the file system. So you are retaining only. The default value here is 60.

Now if you are defining and archive copy group, you can use the parameter RETVer: This is retain version. This specifies the number of days to keep an archive copy. The default value is 365. The possible values used with RETVer are days or no limit.

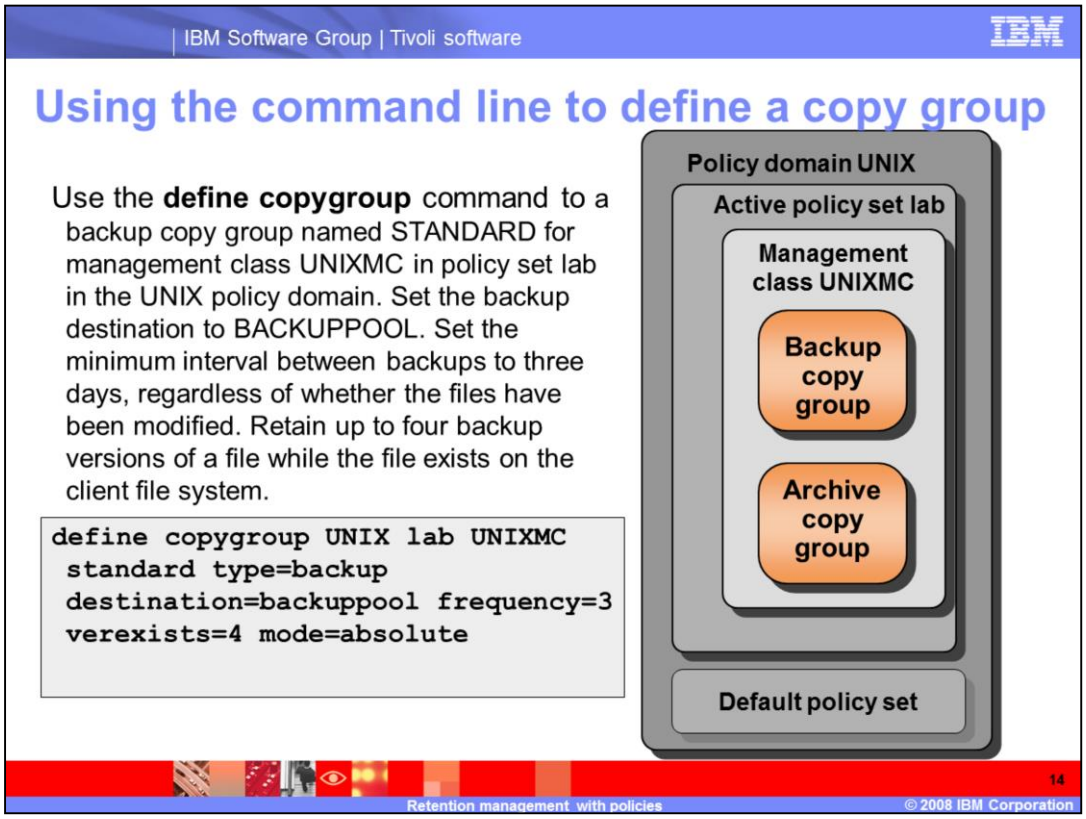
RETInit, or retain initiated. This specifies when the retention time listed with the RETVer attribute is initiated. This parameter is optional, and the default value is creation. The other option would be event.

RETMin. or retain minimum, specifies the minimum number of days to keep an archive copy after it has been archived. The default here would be 365 days.

MODE. Mode will specify whether a file should be backed up based on changes made to the file since the last time it was backed up. This is used only with incremental backups, not with selective. The default value is modified. With archive, the default value is absolute.

SERialization=serialvalue: This specifies how files or directories are handled if they are being backed up or archive modified during backup processing and then what TSM should do if a modification occurs. The default value is SHRSTatic. This means that TSM will

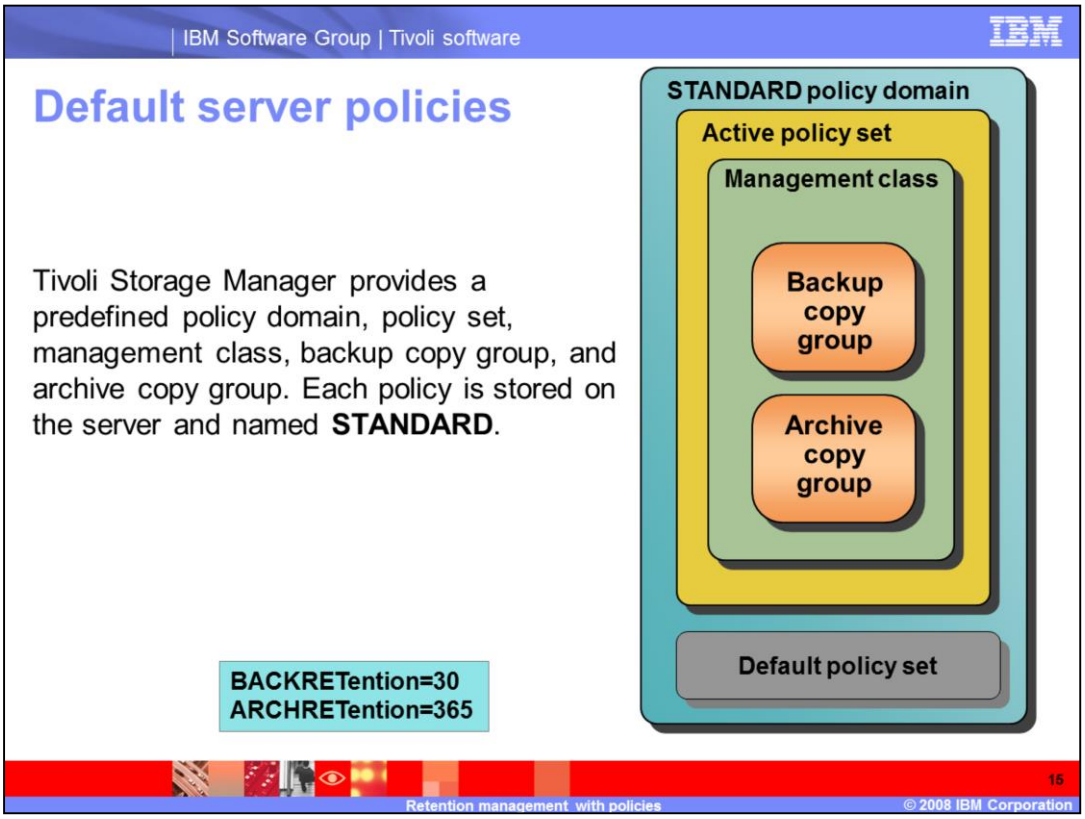
attempt the process up to four times but will not archive or backup a file if it is being modified. With STatic, if it is being modified, the file would not be backed up or archived. Then you have SHRDYnamic and Dynamic which will let you backup or archive files that are being modified. You need to be careful with this because you could get an incomplete backup or archive in this case.



Slide 14 **Using the command line to define a copy group**

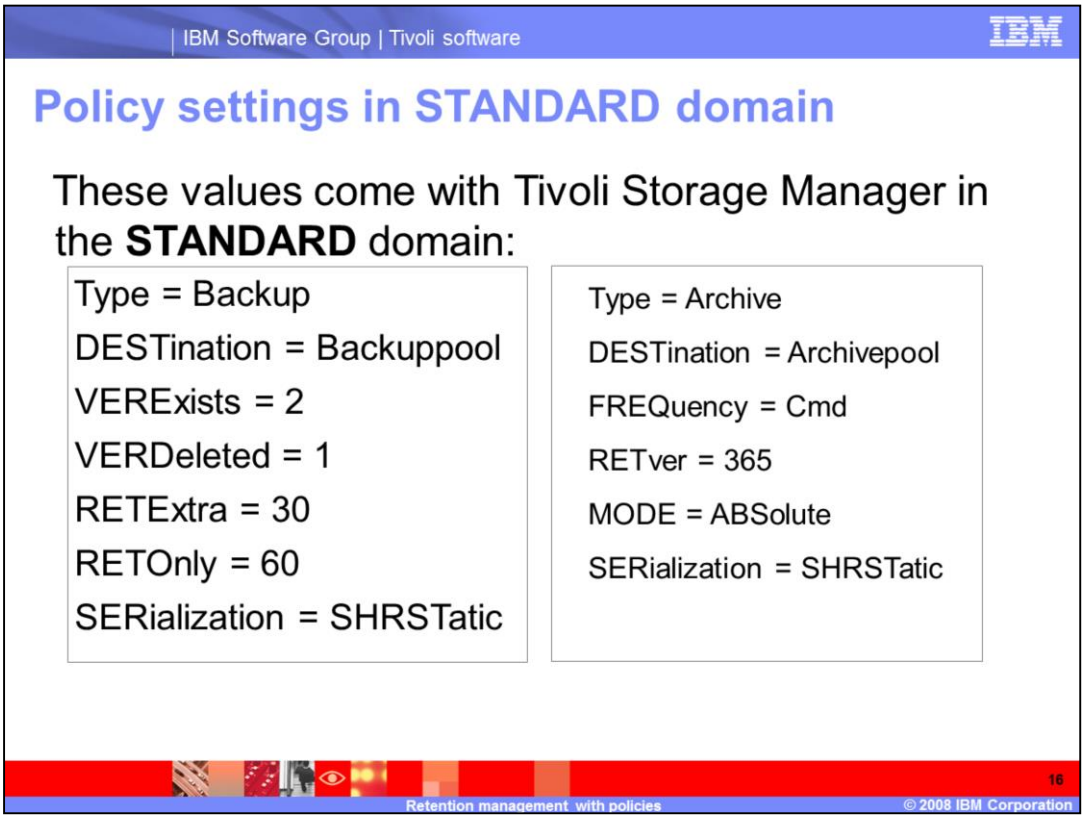
This example is going to define a copy group for a backup copy group named standard, for the management class UNIXMC, in the policy set lab, in the UNIX policy domain. You are going to set the backup destination to the backuppool. Set the minimal interval between backups to three days, regardless if the files have been modified. And retain up to four backup versions of a file while the file still exists on the client file system. The command: **define copygroup UNIX lab UNIXMC standard type=backup destination=backuppool frequency=3 verexist=4 mode=absolute**. Again, here you are using type=backup and standard, even though they are optional (because these are the defaults).

If you were defining a copy group for archive, you would have to mention type=archive.



Slide 15 **Default server policies**

You can begin using TSM immediately with the default policies that are provided. As you become familiar with TSM you are going to want to tailor the standard policy to meet your needs. But as soon as you install TSM, you can run a backup based on these policies, these default policies.

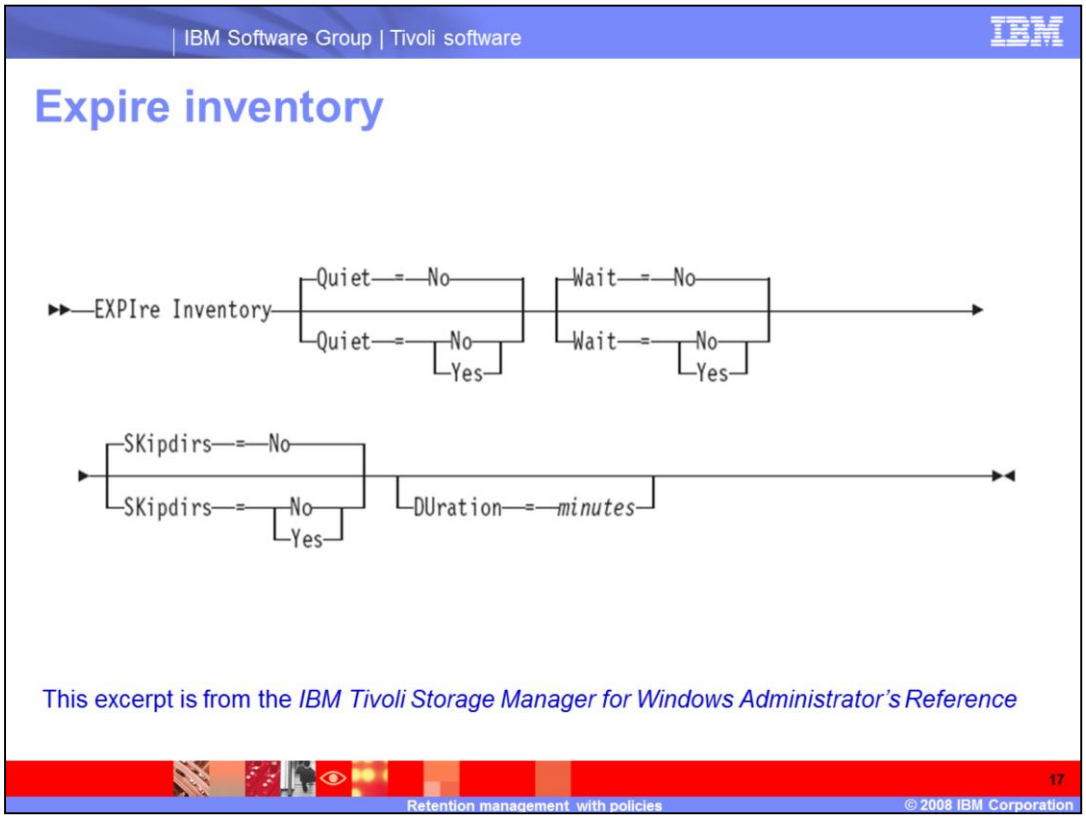


Slide 16 **Policy settings in STANDARD domain**

In the standard domain with TSM, these are the parameters that are preset.

For a backup copy group, the destination is backuppool. Verexists, two versions exist. Verdeleted, one version would be deleted. Retain extra, thirty days and retain only, sixty days. And serialization is set to shrstatic.

For an archive copy group, the destination would be archivepool. Frequency would be command. The retain versions is 365 days. Mode = absolute and serialization = shrstatic.



Slide 17 **Expire inventory**

Expiration is the process by which files are identified for deletion because their expiration date or retention period has passed. Backed up or archived files are marked expired by TSM based on the criteria defined in the backup or archive copy group.

Here you see the command for expire inventory. This manually starts inventory expiration processing. So the inventory expiration process removes client backup and archive file copies from server storage based on policy specified in the copy groups of the management classes to which the files are bound.

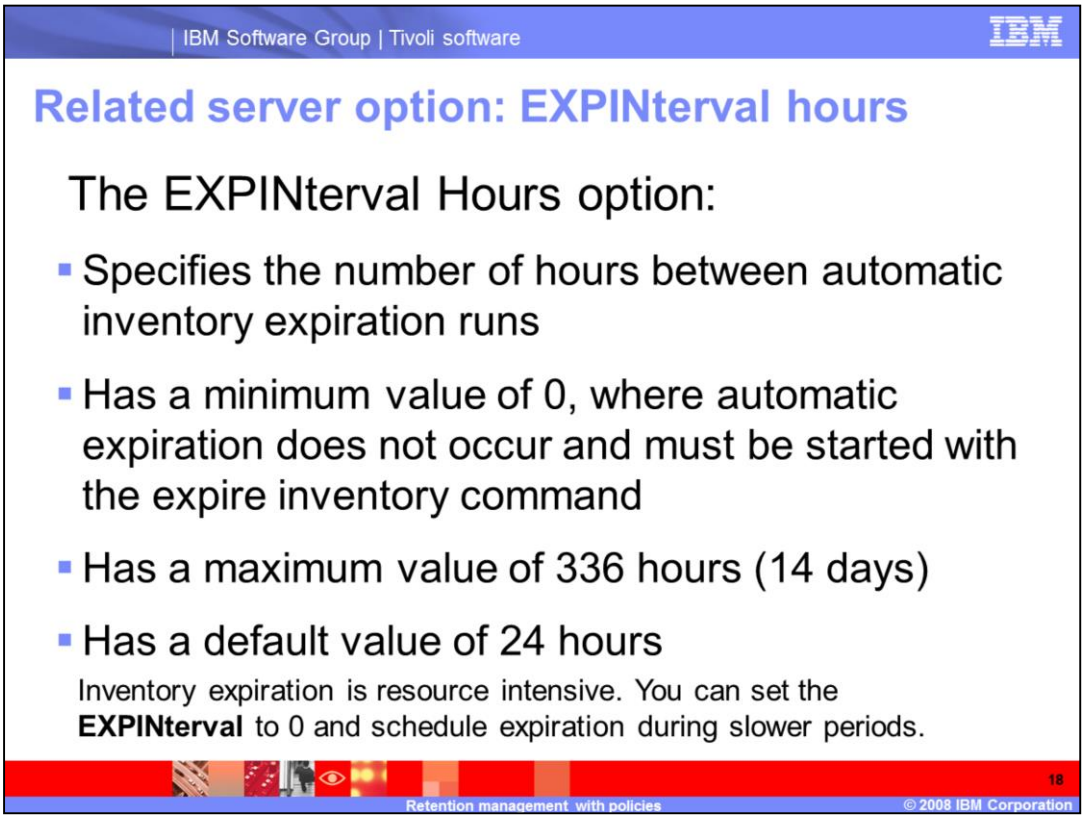
In the command shown here, expire inventory with parameters for Quiet, Wait, Skipdirs, and Duration.

Quite specifies whether the server suppresses detailed messages about policy changes during the expiration processing. The default is no. Another option would be yes.

Wait specifies whether to wait for the server to complete processing this command in the foreground. The default here is no with the option also of yes.

Skipdirs specifies whether the server skips directory type objects during the expiration processing. The default is no. The other possible is of course yes.

And then duration. This specifies the maximum number of minutes for the expiration process to run. And this would be in minutes.



Slide 18 **Related server option: EXPINterval hours**

Copies of files that have expired are not deleted from server storage until expiration

processing occurs. You can run expiration processing either automatically or by issuing the command, expire inventory.

You control automatic expiration processing by using the expiration interval option

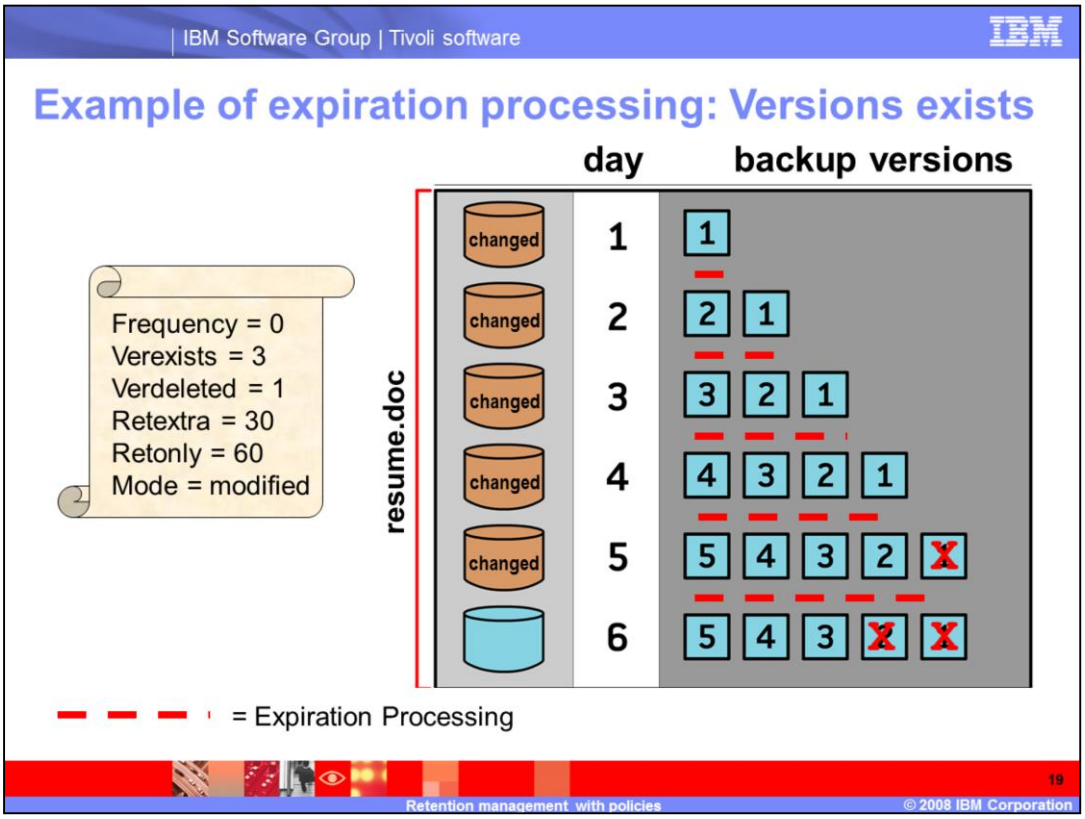
(EXPINTERVAL) in the Tivoli Storage Manager options file (the dsmserv.opt file).

Expiration processing then deletes eligible backup versions and archive copies, based on policies in the backup or archive copy group. By using the expiration intervals option in your dsmserv.opt file, you can specify the number hours between the automatic inventory expiration processes. There is a minimum value of zero, where expiration does not occur automatically and must be started by using the expire inventory command. Maximum value of 336 hours, which is 14 days, and the default value of 24 hours.

Now remember the process is a bit resource intensive. If you want, you can schedule the expire inventory command at slower periods.

If for any reason you need to cancel inventory expiration, use the command **cancel expiration**.

Do not cancel the process.



Slide 19 **Example of expiration processing: Versions exists**

The slide shows the life cycle of a backed up client file as it is stored in a TSM storage pool.

A scheduled incremental backup is run every evening, followed by expire inventory.

The copy group values for this file are: Frequency = 0. Versions exists = 3 (which is the Maximum). Versions deleted = 1. Retain extra = 30 days (this is if the file still exists on workstation). Retain only = 60 days (This is for files that have been deleted from workstation). And mode = modified.

On day 1 the file is created. That evening it is backed up and then expiration runs. The file would be marked as active, and expiration has nothing to expire.

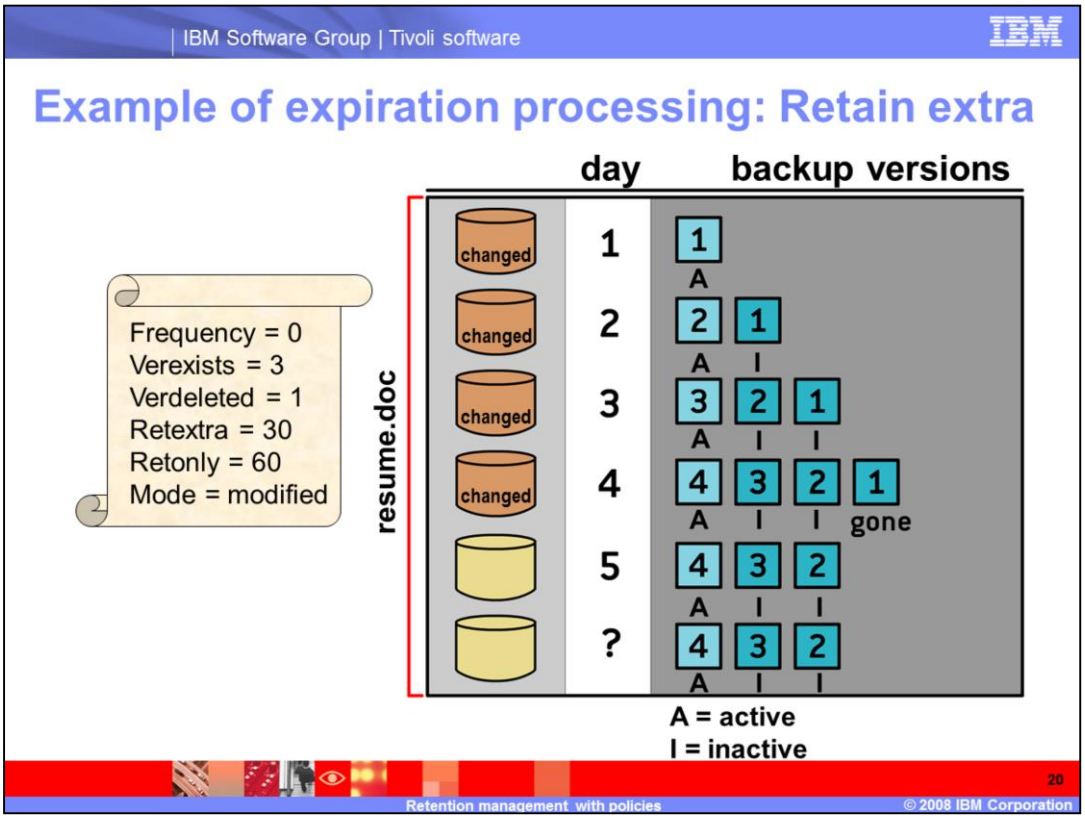
On day 2, the file is changed, a backup is run, and expiration is run. The backup for day 2 is now the active file. The backup from day 1 becomes inactive.

On day 3, the file is changed, backed up, and expiration runs. The backup for day 3 is now the active version. The backups from days 1 and 2 are inactive. Up to this point expiration has not removed any files, because there are three versions that must exist.

So, on day 4, when the file is changed, backed up, and expiration runs, the file for day 4 is now the active file, and the file from day 1 is removed during expiration processing.

On day 5, the file is changed, backed up, and expiration is run. The file from day 5 is now the active file. And again, because you are only keeping three versions of the file, the backup from day 2 also expires and is removed during expiration processing.

On day 6, the file is not changed, therefore it is not backed up. Because there are three versions of the file after expiration on day 5, there is no change during expiration processing. The file from day 5 is active and the files from days 4 and 3 are inactive. There are three versions of the file.



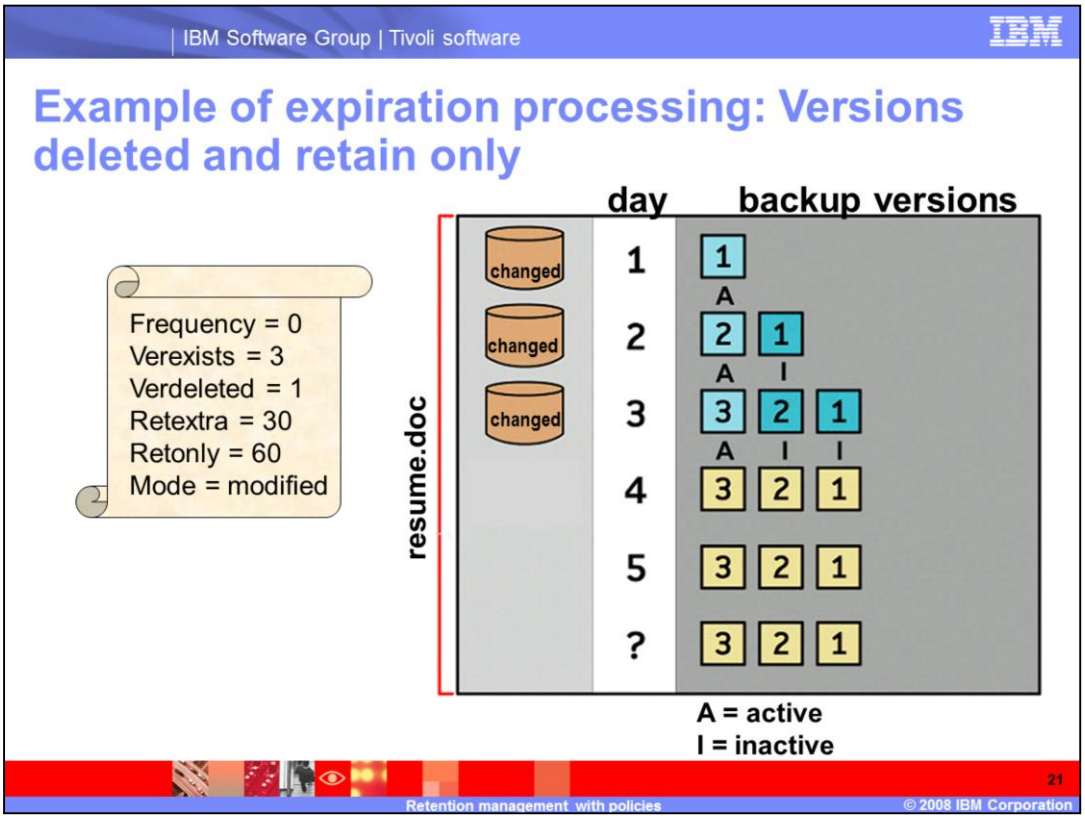
Slide 20 **Example of expiration processing: Retain extra**

Again in this scenario, you have the same parameters for the copy group, and a scheduled incremental backup is run every evening, followed by expire inventory.

Note that backup 1 on day 4 is now gone because you are only keeping only three versions based on VERExist=3. Also in this scenario, the file goes unchanged after day 4. In this case, can you predict the results for day 33?

On day 33, only one version remains. This is because retain extra is set to 30. if you always want 3 versions retain extra needs to be no limit.

Can you predict when backup 2 would expire? Backup 2 expires on day 33 because it became inactive on day 3 and you would add 30 days based on RETExtra=30.



Slide 21 **Example of expiration processing: Versions deleted and retain only**

Again, you are using the same copy group parameters as previously and a scheduled incremental backup is run every evening, followed by expire inventory.

What happens if the file is removed from the client workstation on the morning of day 4?

Backup 3 will become inactive when the original file is deleted.

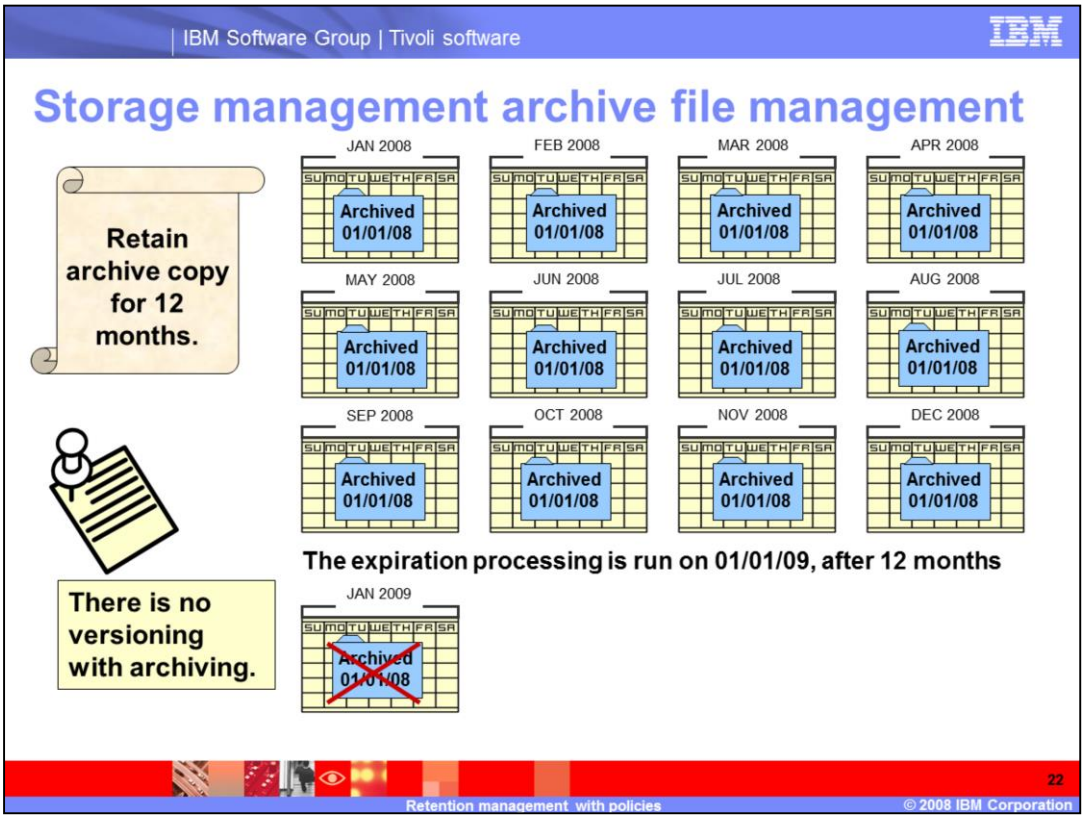
How many extra copies will be there after the incremental backup on day 4?

To determine this, notice that VERDeleted is set to 1. This means that

backups 1 and 2 will be removed, because once the original file has been removed from the client's workstation, one file will remain in storage.

Can you predict when the only backup will expire and be removed?

The only backup will expire and be removed on day 64. This because RETonly is set to 60 days.

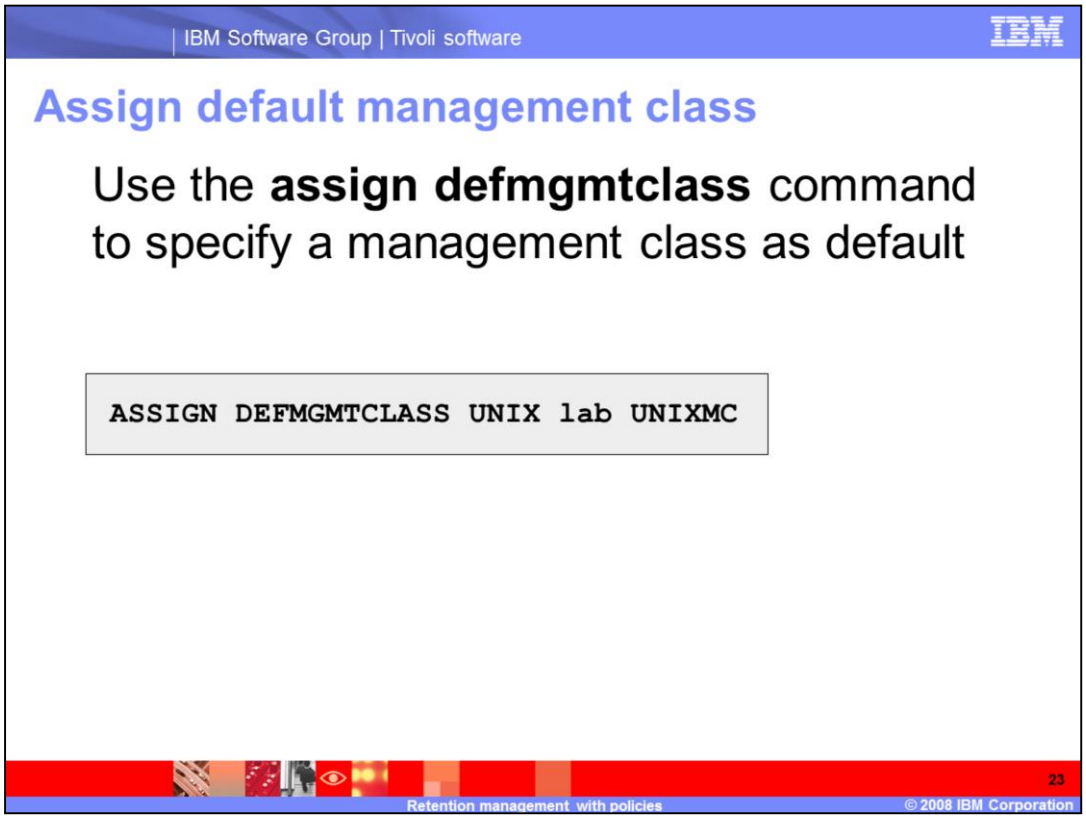


Slide 22 **Storage management archive file management**

The retain version that is specified in the archive copy group specifies the number of days an archived copy remains in data storage. When the specified number of days elapses for the archived copy of the file, IBM Tivoli Storage Manager deletes the file from data storage.

Unlike backup, archive does not allow multiple versions of data files and directories.

So you see here in this example, a file that was archived on January 1, 2008 and the policy is to retain the file for twelve months. On the thirteenth month that file would be deleted.

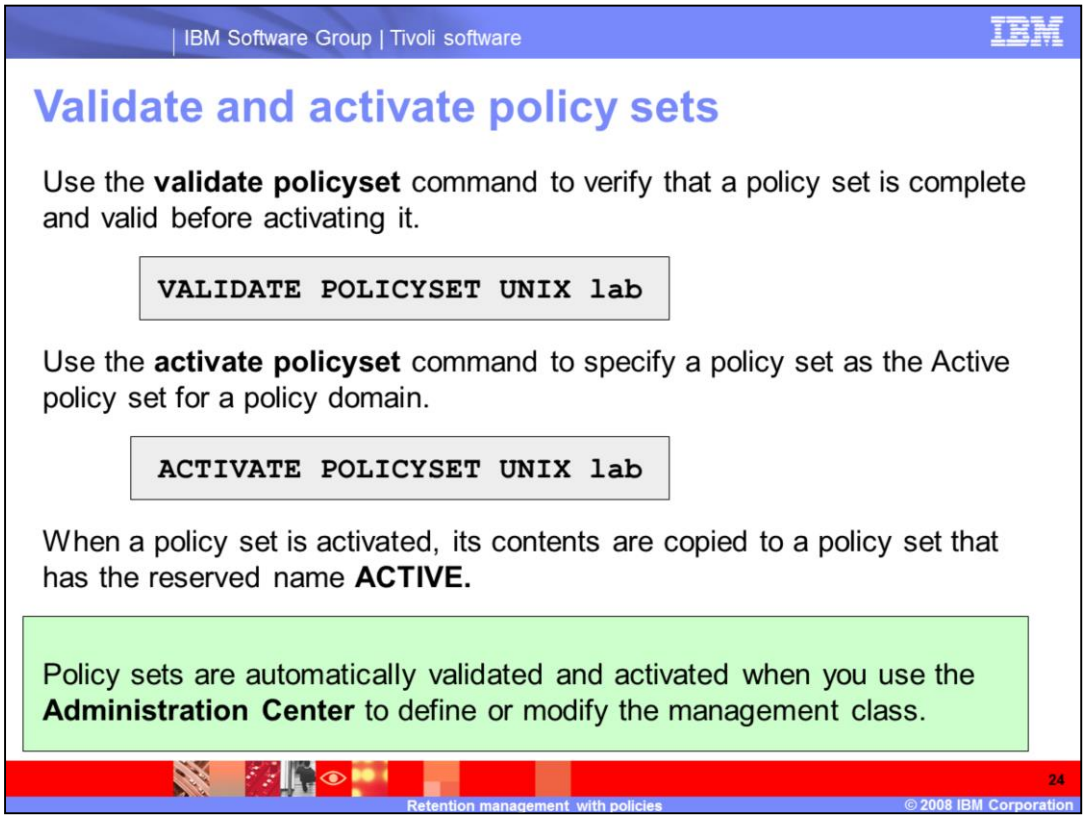


Slide 23 **Assign default management class**

You must assign a default management class for a policy set before you can activate that policy set. To ensure clients can always back up and archive files, choose a default management class that contains both an archive copy group and a backup copy group.

You will be specifying the domain name, set name, and the class name.

And here you see an example of the command, **assign defmgmtclass UNIX lab UNIXMC**. I think when they wrote the commands they were not really expecting people to pronounce them. Anyway, you can see the command here on the slide.



Slide 24 **Validate and activate policy sets**

Remember, before you can use a policy set, it needs to be validated then activated.

The validate command examines the management class and copy group definitions for a

specified policy set and reports on conditions that need to be considered if the policy set is to be activated. So, after a change is made to a policy set, and the policy set is validated, then the policy set must be activated to make it the ACTIVE policy set.

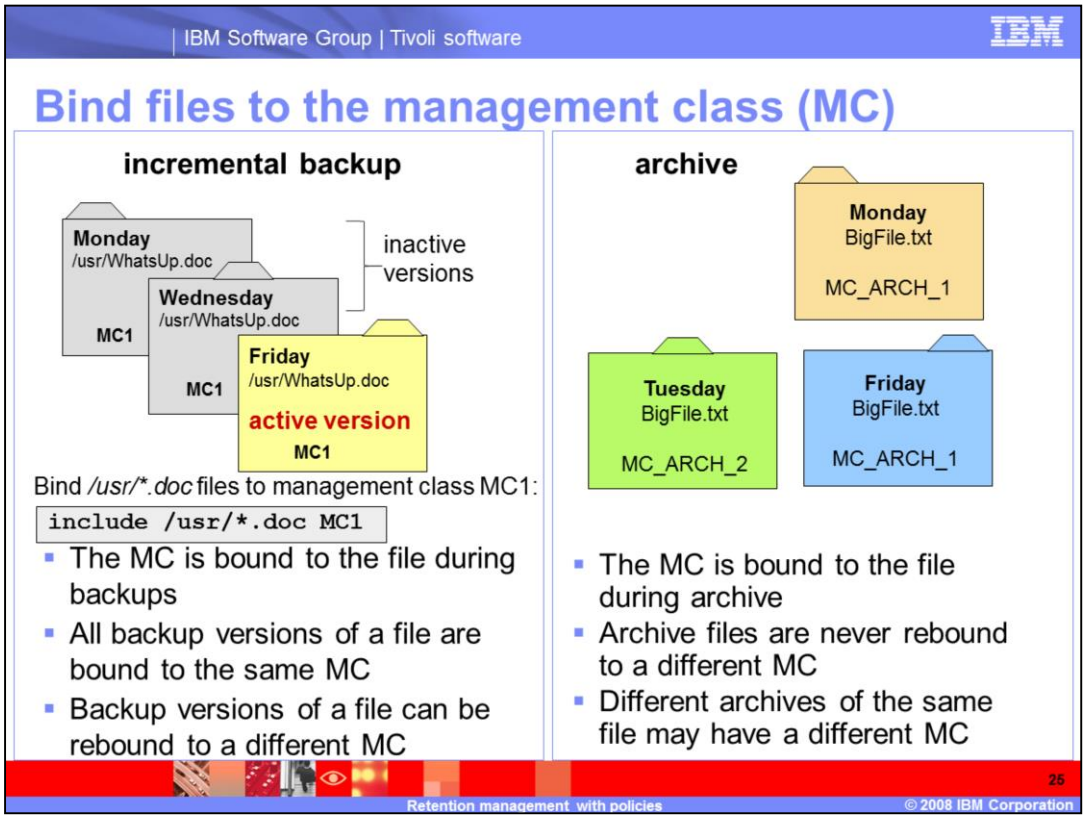
The validate policy set command will fail if any of the following conditions exist:

First, a default management class is not defined for the policy set.

Or a copy group within the policy set specifies a copy storage pool as a destination.

Or if a management class specifies a copy pool as the destination for space-managed files.

With Tivoli Storage Manager version 5.5, if you use the Administration Center to define or modify the management class, policy sets are automatically validated and activated.



Slide 25 **Bind files to the management class**

Binding is the process of associating a file name with a management class name. To bind, you can specify the management class name using the INCLUDE option on an include-exclude list.

By using the ARCHMC option when archiving a file.

Or you can bind a directory to a management class using the DIRMC option when backing up a file.

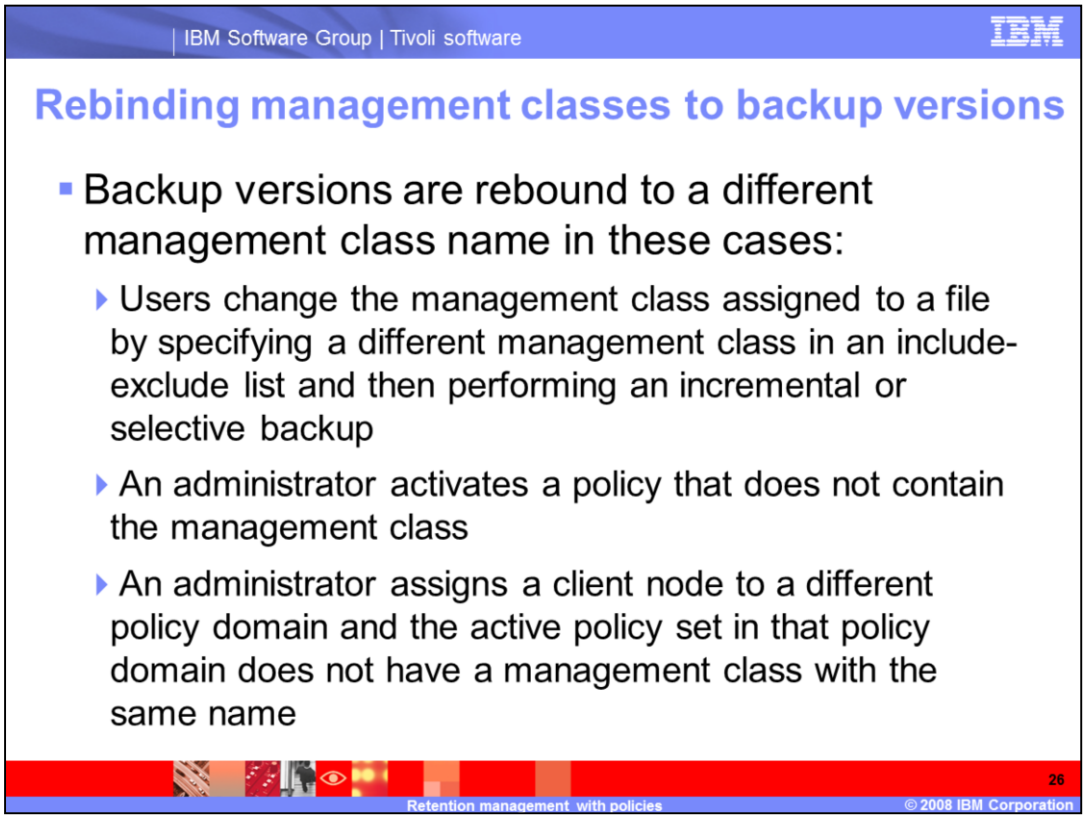
A client node can bind a file to the default management class in the active policy set, when you do not bind to a specific management class name.

Rebinding is the process of associating a file with a new management class name.

Archive files are never rebound since each archive operation creates a different archive copy.

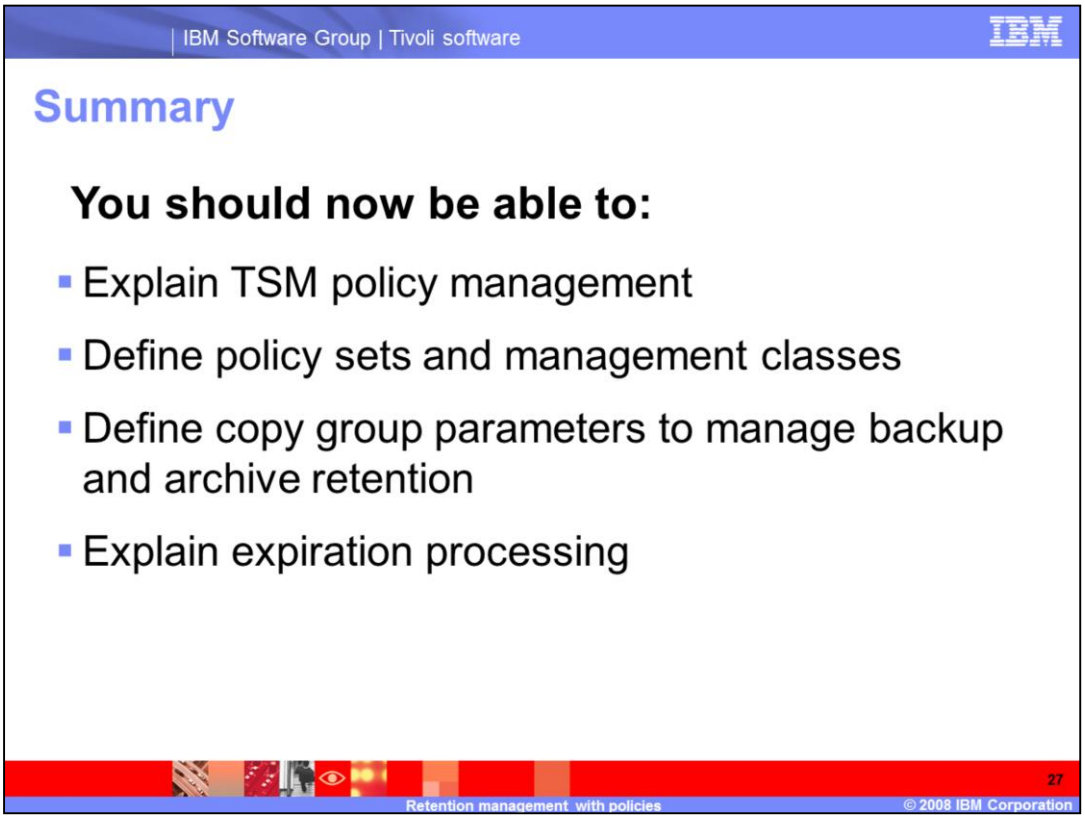
The example you see here on the left is if you want to bind all the documents in the usr directory to a management class called MC1. So you would add the include /usr/*.doc MC1 to the include-exclude list. Then, the management class is bound to the files during backups, based on the include-exclude statement. All backup versions of a file are bound to the same management class. But backup versions of a file can be rebound to a different management class

With an archive, the management class is bound to the file during archive, archive files are never rebound to a different management class, and different archives of the same file may have a different management class.



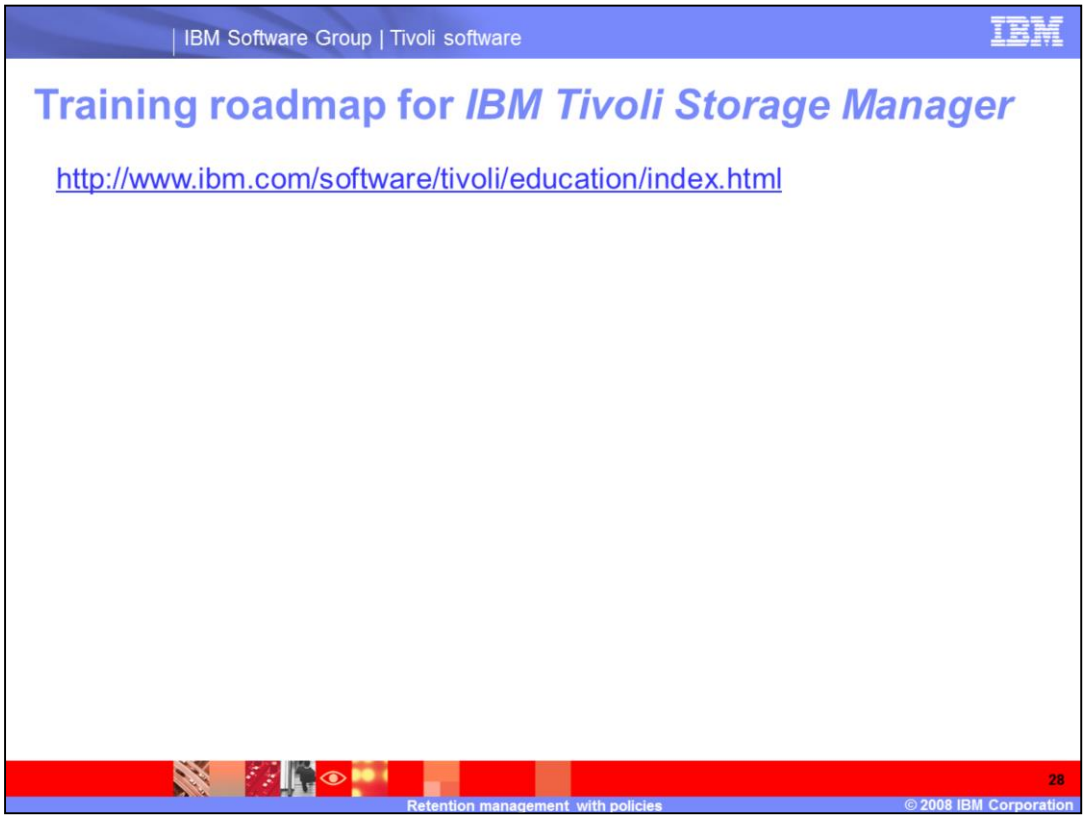
Slide 26 **Rebinding management classes to backup versions**

Backup versions can be re-bound to a different management class name, if you change the management class assigned to a file by specifying a different management class in an include-exclude list and then perform a backup. Or, an administrator activates a policy that does not contain the management class. Or an administrator assigns a client node to a different policy domain and the active policy set in that policy domain does not have a management class with the same name.



Slide 27 **Summary**

You should now be able to explain TSM policy management, define policy sets and management classes, define copy group parameters to manage backup and archive retention, and explain expiration processing



Slide 28 **Training roadmap for *IBM Tivoli Storage Manager***

If you go to http://www.ibm.com/software/tivoli/education/edu_prd.html this will take you to the training page for Tivoli Storage Manager version 5.5.

Slide 29 (Copyright)

This concludes the IBM Education Assistant training for IBM Tivoli Storage Manager version 5.5 managing backup-archive retention values with TSM policies.