SD-UX Depot Management Tools and Tricks

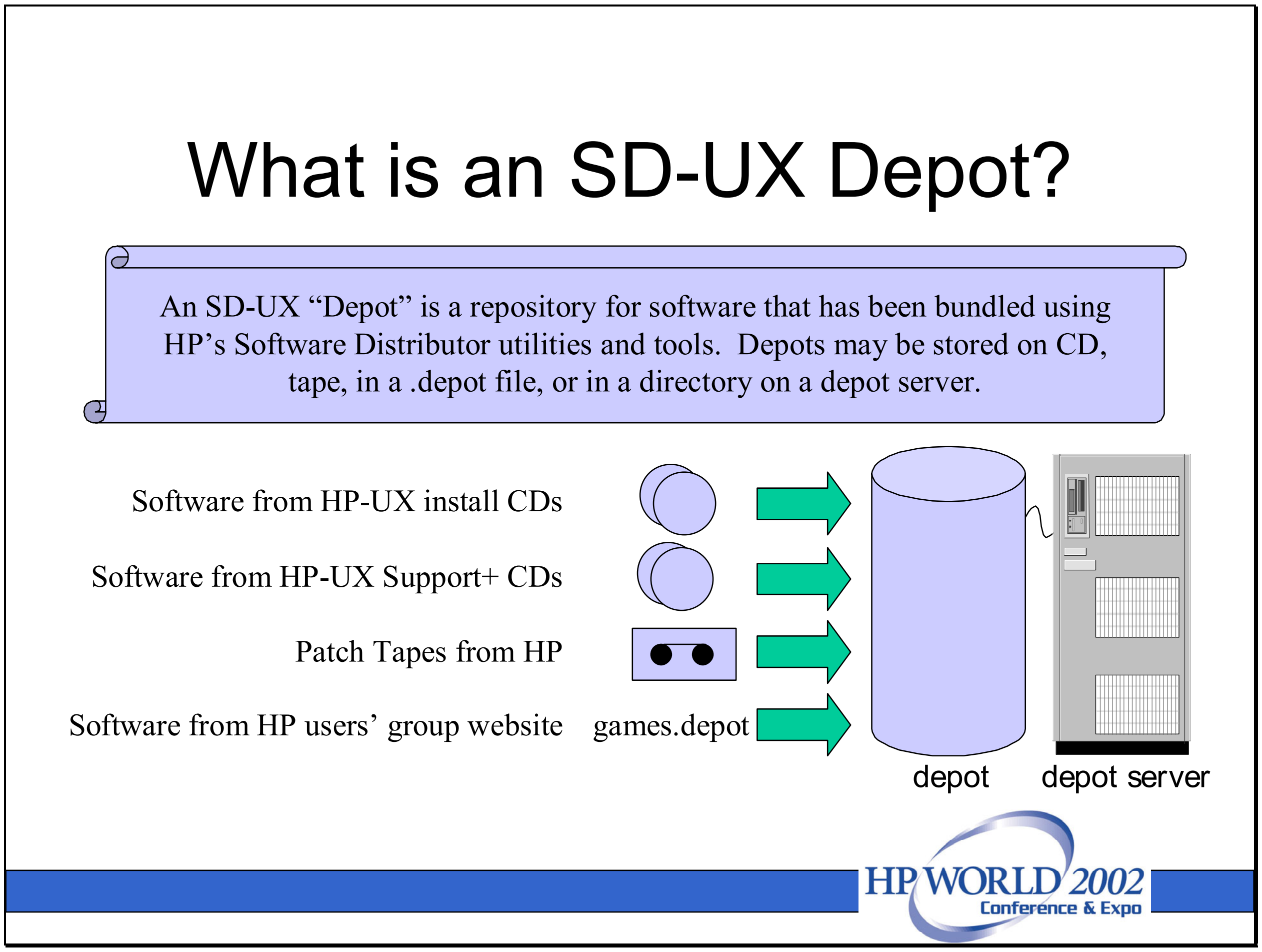
Darren Miller
Senior Education Consultant
HP Customer Education
Darren_Miller@hp.com

HP WORLD 2002 Conference & Expo

Managing software in today's large computing environments can be a challenging task. Administrators often manage dozens of systems, and must contend with a constant stream of software and patch updates.

Fortunately, all software from HP – from the HP-UX install CD's, to Openview product CDs, to patch downloads from the ITRC -- are packaged using HP's Software Distributor UX (SD-UX) utilities. The SD-UX utilities make it fairly easy to install, remove, and catalog software installed on a system.

Administrators that manage multiple systems can streamline software management even further by taking advantage of SD-UX software depots. This paper will explore some of the tools and techniques available to simplify software and depot management in HP-UX 11.x.



What is an SD-UX Depot?

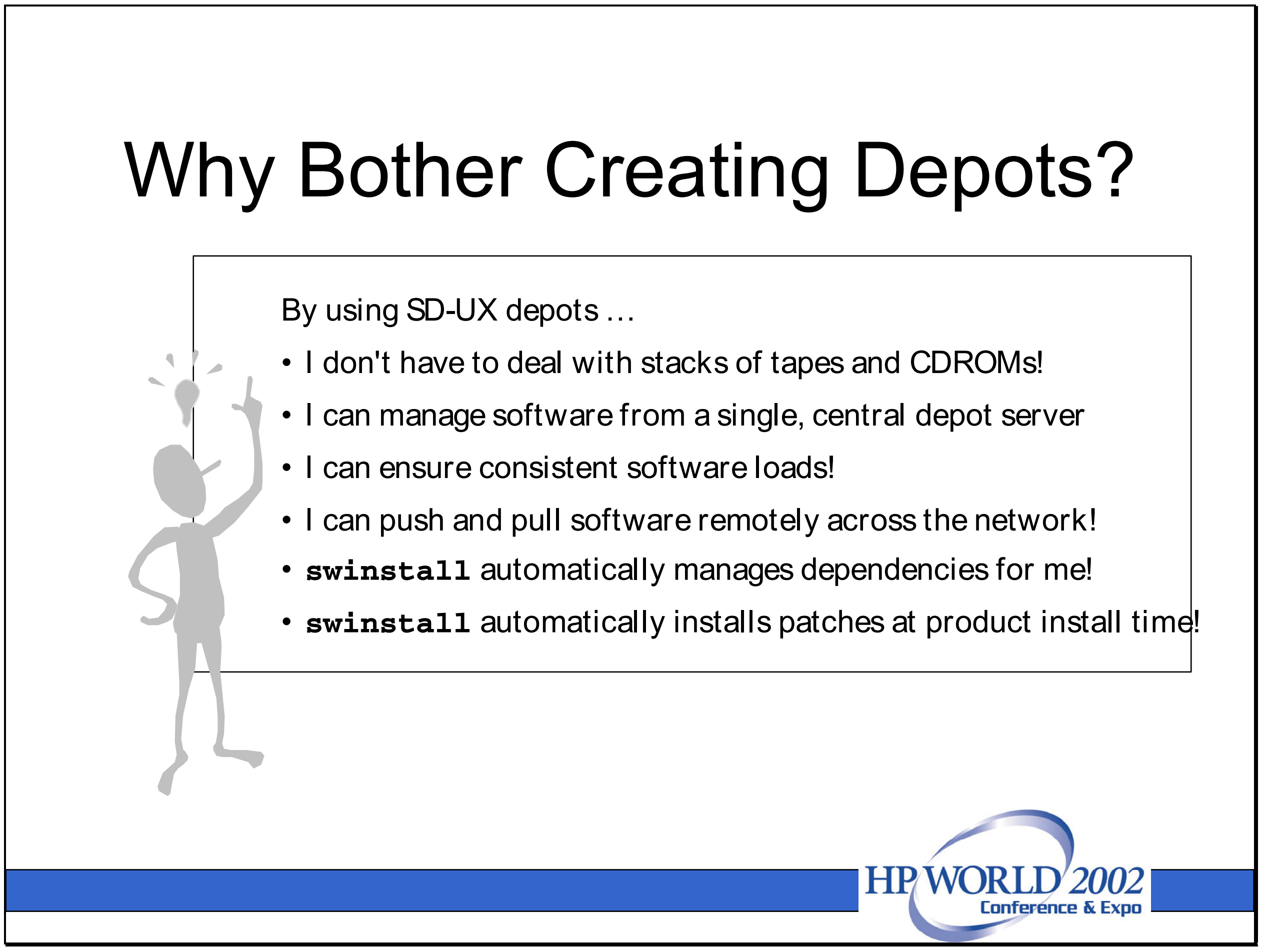
An SD-UX “Depot” is a repository for software that has been bundled using HP’s Software Distributor utilities and tools. Depots may be stored on CD, tape, in a .depot file, or in a directory on a depot server.

Software from HP-UX install CDs
Software from HP-UX Support+ CDs
Patch Tapes from HP
Software from HP users’ group website
games.depot
depot
depot server
HP WORLD 2002 Conference & Expo

Perhaps we should start by defining “depot”. An SD-UX depot is a repository for software packaged using the SD-UX utilities. Depots can be stored using a variety of media.

- The OS, application, and Support+ software that you receive in the HP-UX media kit are structured as CDROM depots.
- The Support+ patch bundles that are distributed several times each year are structured as CDROM depots, too.
- Patches that you download from the ITRC website are stored as **`.depot`** files. Contributed software that you download from the HP users’ group typically comes in a **`.depot`** file, too.
- Occasionally, HP support personnel may provide a patch tape, which is also recorded in the SD-UX depot format.

Juggling stacks of media kits, CDROMs, tapes, and **`.depot`** files can be challenging. Fortunately, SD-UX offers a better solution: using the **`swcopy`** command, you can consolidate software from multiple sources in one or more directory depots on an SD-UX depot server. Any host on your network can then install software directly from the depot server.



Why Bother Creating Depots?

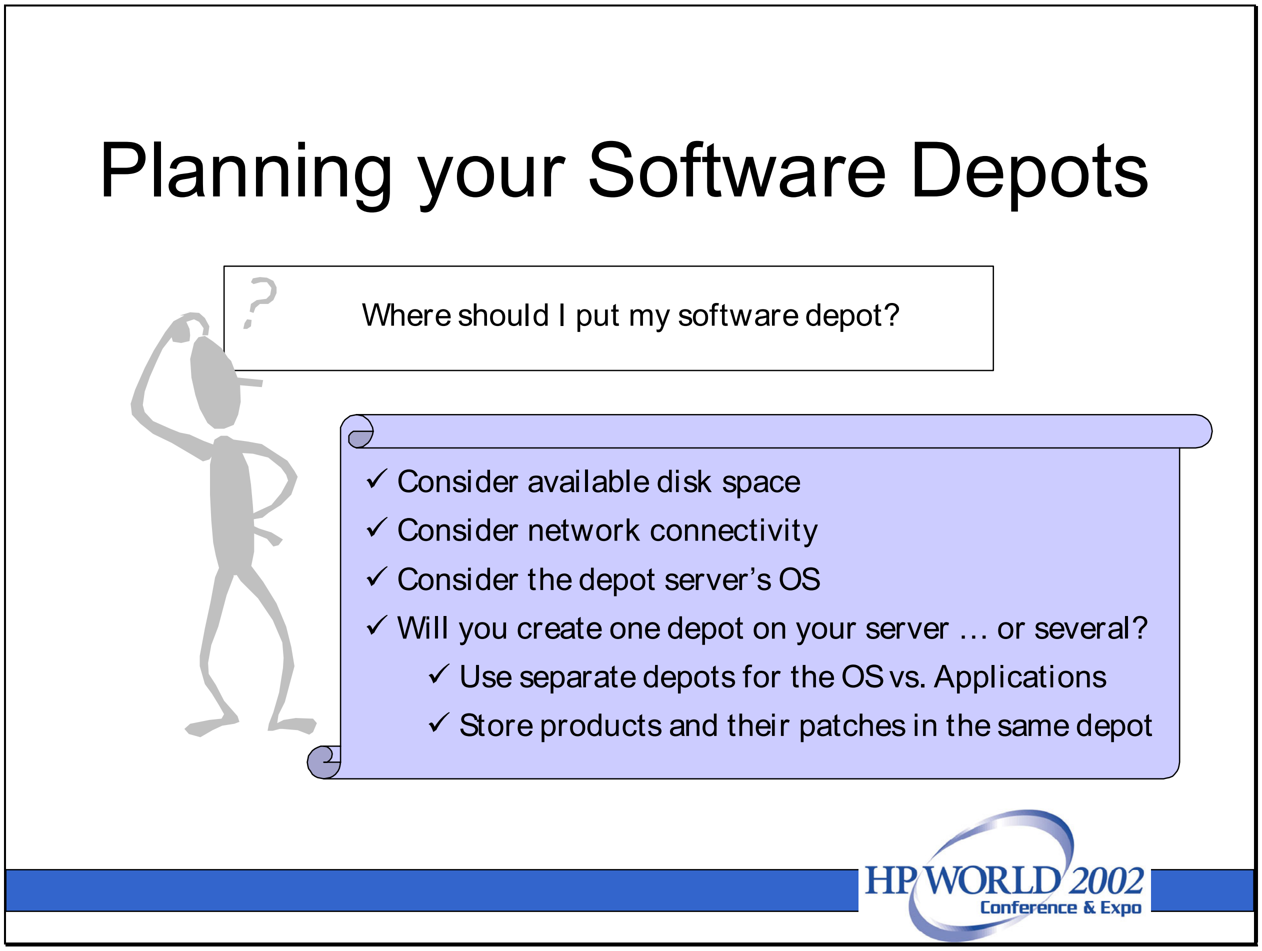
By using SD-UX depots …

- I don't have to deal with stacks of tapes and CDROMs!
- I can manage software from a single, central depot server
- I can ensure consistent software loads!
- I can push and pull software remotely across the network!
- **`swinstall`** automatically manages dependencies for me!
- **`swinstall`** automatically installs patches at product install time!

HP WORLD 2002 Conference & Expo

As noted on the slide above, consolidating software in an SD-UX depot offers many advantages:

- Instead of managing stacks of CDROMs and tapes, you can **`swinstall`** software and patches from your SD-UX depot server. This is especially helpful when installing systems that don't have a CDROM or tape drive available.
- A depot server provides a single point of administration for your software and patch updates.
- Installing all of your hosts from a central depot server ensures that all hosts have a similar software/patch image.
- Configuring a depot server makes it possible to remotely install and manage software. Individual hosts on your network can "pull" software from the depot server. With HP-UX 11i, it is now possible to "push" software installs and updates from a depot server to one or more remote targets.
- After you select a patch, product, or bundle in a depot, **`swinstall`** auto-selects other products from the depot that your selected product requires.
- When **`swinstall'`**ing software from an SD-UX depot, if the depot contains patches for the user-selected product(s), **`swinstall`** will automatically select and install those patches at the same time that it installs the selected product itself. This can significantly decrease the amount of downtime required to update software and patches on a system.



Planning your Software Depots

Where should I put my software depot?

- ✓ Consider available disk space
- ✓ Consider network connectivity
- ✓ Consider the depot server's OS
- ✓ Will you create one depot on your server … or several?
  - ✓ Use separate depots for the OS vs. Applications
  - ✓ Store products and their patches in the same depot

HP WORLD 2002 Conference & Expo

Several important design issues should be considered before you configure an SD-UX server.

Consider available disk space

Each depot is configured as a directory tree. The more software you intend to store in your depot, the more disk space you will need. To simplify disk space management, it may make sense to create a separate file system for your SD-UX depots. The commands below might be used to create a 2GB depot file system in the vgxx volume group.

```
# vgdisplay vgxx
# lvcreate -L 100 -n depots vgxx
# newfs -F vxfs /dev/vgxx/rdepots
# mkdir /depots
# mount /dev/vgxx/depots /depots
# vi /etc/fstab
```

Consider network connectivity

Installing software remotely from a depot server generates a fair amount of network traffic. Ensure that your depot server has adequate network bandwidth.

It may not be feasible to push or pull software installs across slow WAN links. If you need to install software on systems in branch offices with poor connectivity, it may make sense to create replica depot servers, or distribute custom depot tapes. This will be described later in the presentation.

Consider the depot server's OS

If your depot server will serve multiple clients running different versions of HP-UX, ensure that the most recent version of SD-UX is loaded on the server. An 11i depot server can host 11i depots, as well as 11.00 and 10.20 depots for clients running earlier versions of HP-UX. Forward compatibility, however, cannot be guaranteed with earlier versions of SD-UX.

Will you create one depot on your server … or several?

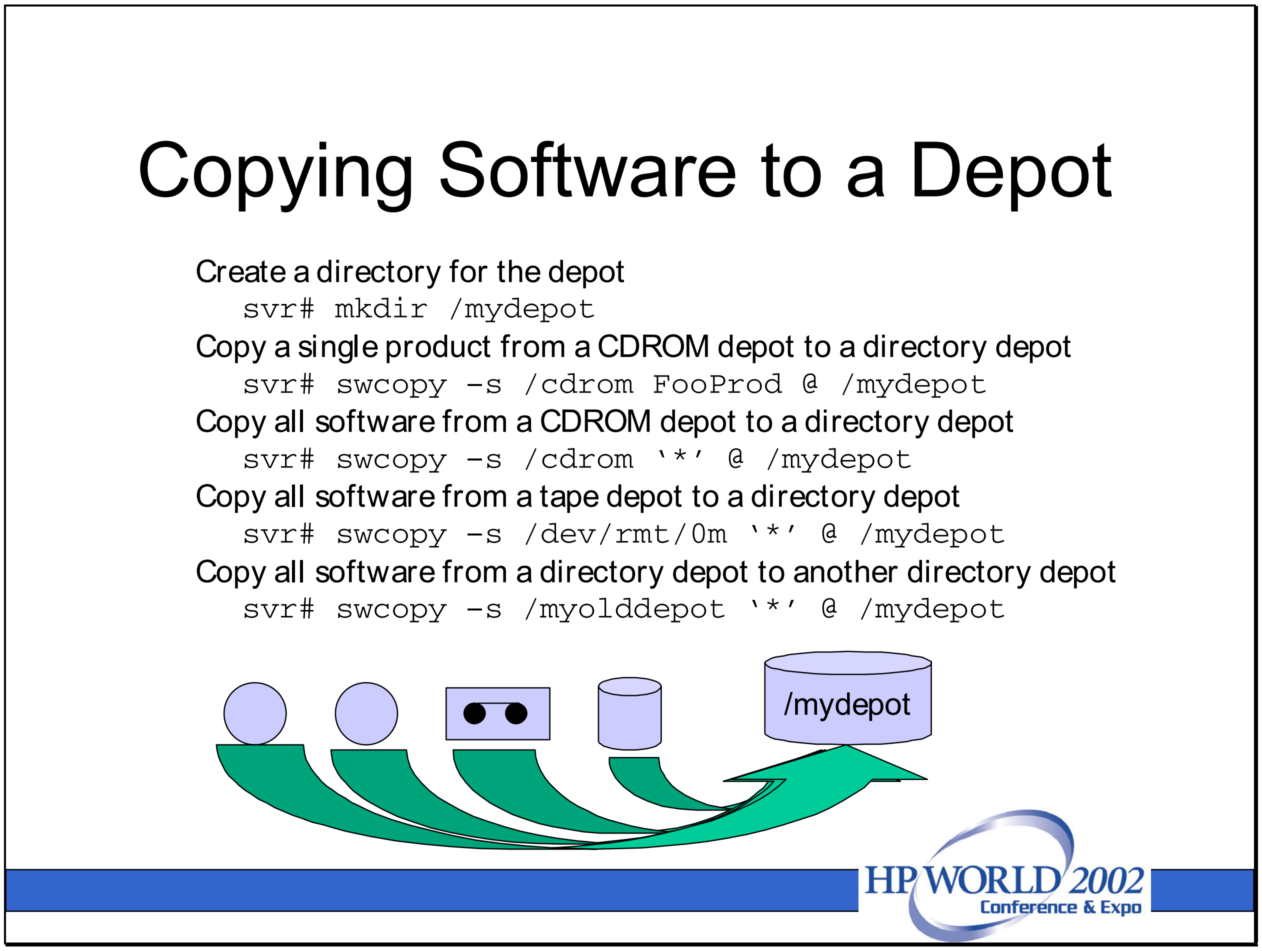
A single depot server can host multiple software depots. Ideally, you should:

- Create a separate depot for each version of the OS
- Separate application from OS software

Each depot requires a separate directory:

```
/depots/11i/MissionCriticalOE
/depots/11i/TechnicalComputingOE
/depots/11i/Applications
```

Prior to HP-UX 11.00, patches and products had to be stored in separate depots. HP-UX 11.00 introduced some **`swinstall`** enhancements that made it practical to co-mingle patches and products in a single depot.



Copying Software to a Depot

Create a directory for the depot

```
svr# mkdir /mydepot
```

Copy a single product from a CDROM depot to a directory depot

```
svr# swcopy -s /cdrom FooProd @ /mydepot
```

Copy all software from a CDROM depot to a directory depot

```
svr# swcopy -s /cdrom '*' @ /mydepot
```

Copy all software from a tape depot to a directory depot

```
svr# swcopy -s /dev/rmt/0m '*' @ /mydepot
```

Copy all software from a directory depot to another directory depot

```
svr# swcopy -s /myolddepot '*' @ /mydepot
```

Once you have decided where to put your depot, create a directory for the depot, and copy the desired software to it using the **swcopy** command. As shown on the slide, **swcopy** may be used to copy all or selected software from CD depots, tape depots, or other directory depots. Simply specify the source with **swcopy -s**!

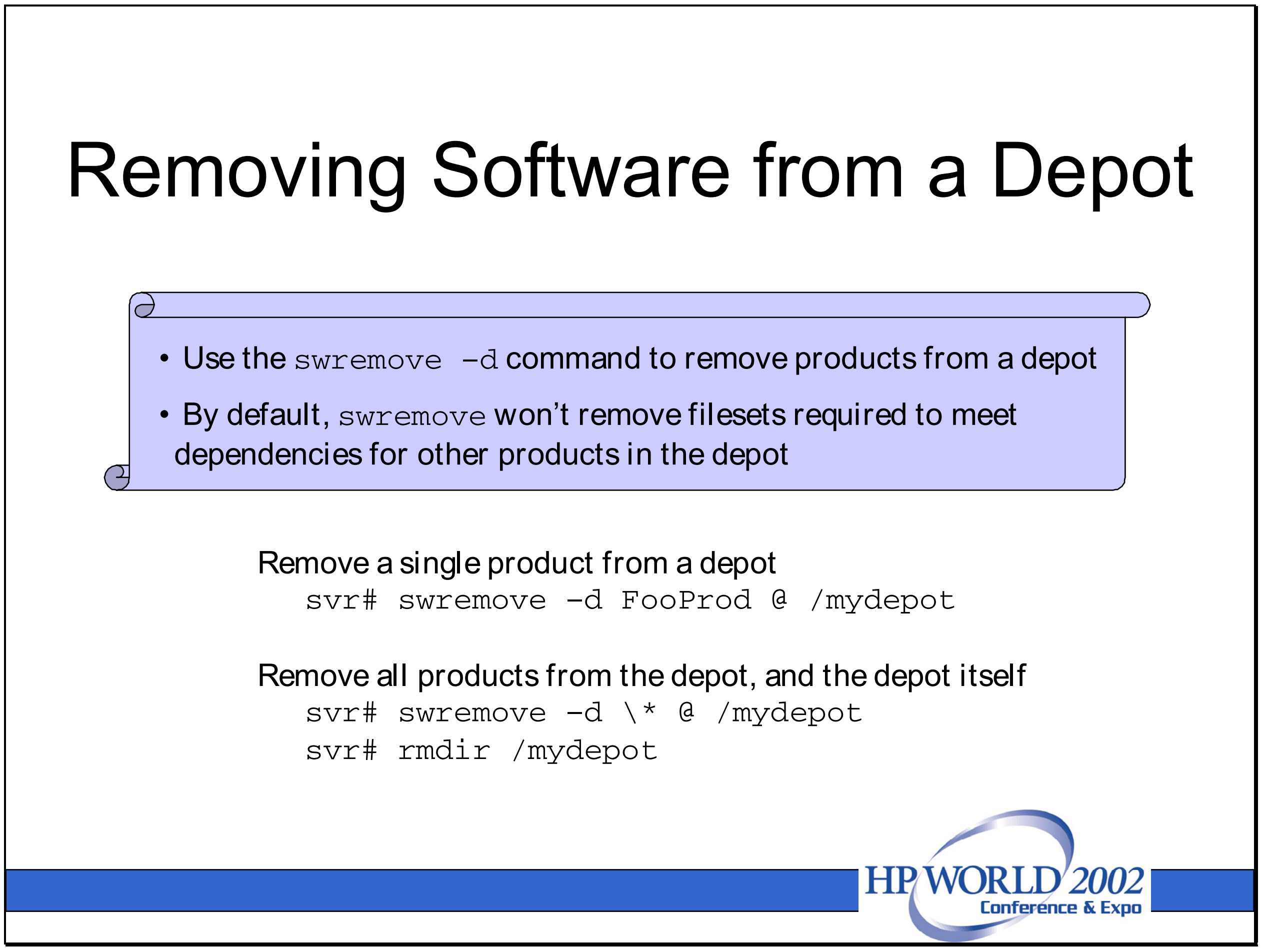
Dealing with dependencies

Some filesets require one or more additional filesets in order to satisfy software dependencies. By default, **swcopy** enforces these dependencies when copying software to a depot. Some filesets may be automatically selected for you to meet dependencies. If dependencies can't be met, **swcopy** will fail.

When setting up a patch depot it is common to have a collection of downloaded **.depot** files which contain only the single, individual patches themselves. In such situations, it will be necessary to override dependency checking to setup the depot.

```
# swcopy -s /tmp/PHCO_1000.depot \
         -x enforce_dependencies=false \* @ /mydepot
```

You should never, however, override dependency checking when **`swinstall`**'ing software from the depot. Doing so may leave your system in an inconsistent state.



Removing Software from a Depot

- Use the `swremove -d` command to remove products from a depot
- By default, `swremove` won't remove filesets required to meet dependencies for other products in the depot

Remove a single product from a depot

```
   svr# swremove -d FooProd @ /mydepot
```

Remove all products from the depot, and the depot itself

```
   svr# swremove -d \* @ /mydepot
   svr# rmdir /mydepot
```

HP WORLD 2002 Conference & Expo

The command required to remove a product from a depot is fairly straightforward:

```
  svr# swremove -d FooProd @ /mydepot
```

If you wish to remove all of the software from a depot, simply replace **`FooProd`** with a **`'*'`**. This will also make the depot itself inaccessible to clients..

```
  svr# swremove -d '*' @ /mydepot
```

Dealing with dependencies

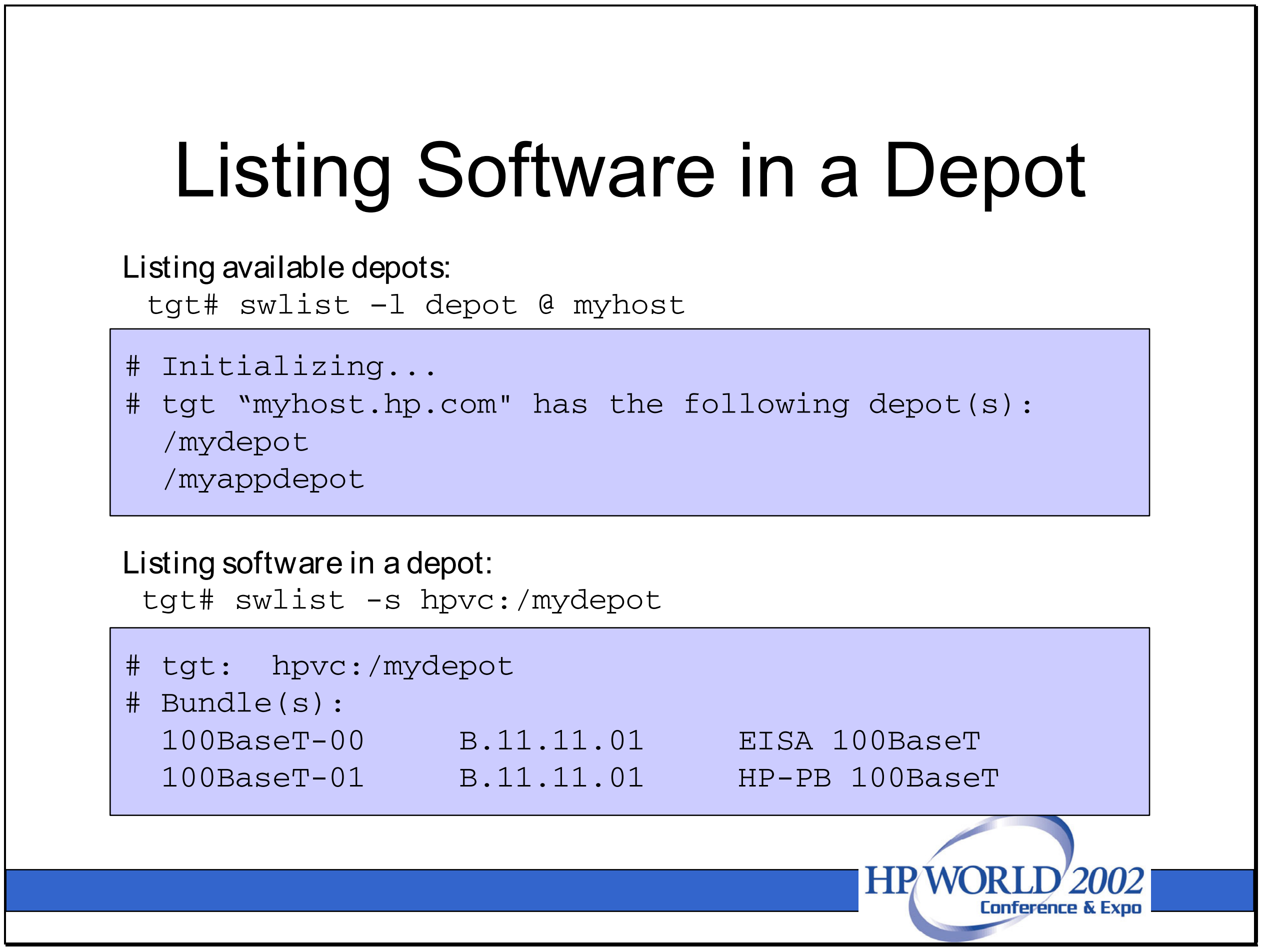
Sometimes other products or patches in your depot may be dependent on the product you wish to remove. In this situation, removing the product or patch from the depot becomes more complicated. Two **`swremove`** options control what happens.

The **`-x enforce_dependencies=true|false`** option determines whether **`swremove`** allows a patch or product to be removed, if that patch or product is required by other patches or products. The default value for this option is "true".

The **`-x autoselect_dependents=true|false`** option determines whether **`swremove`** selects just the explicitly selected patch, or the explicitly selected patch and all of its dependents. The default value for this option is "false".

The table below summarizes the resulting **`swremove`** behavior you will see when using the most common combinations of these options to remove a patch that has dependencies:

`enforce_dependencies`	**`autoselect_dependents`**	**result**
true	false	nothing removed (default)
false	false	patch removed, dependents remain
true	true	patch and dependents removed



Listing Software in a Depot

Listing available depots:

```
 tgt# swlist -l depot @ myhost
```

```
# Initializing...
# tgt "myhost.hp.com" has the following depot(s):
  /mydepot
  /myappdepot
```

Listing software in a depot:

```
 tgt# swlist -s hpvc:/mydepot
```

```
# tgt:  hpvc:/mydepot
# Bundle(s):
  100BaseT-00     B.11.11.01     EISA 100BaseT
  100BaseT-01     B.11.11.01     HP-PB 100BaseT
```

HP WORLD 2002 Conference & Expo

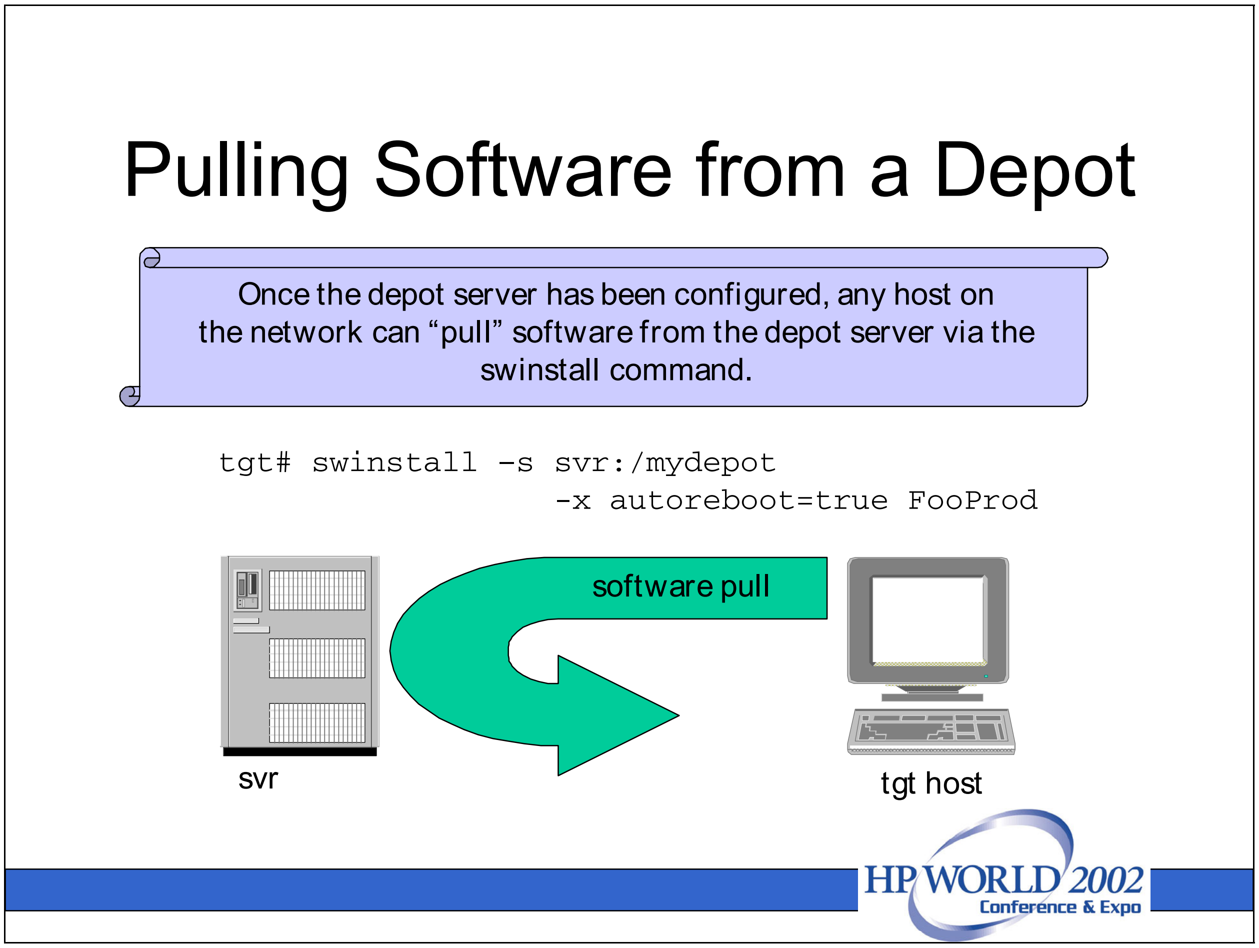
Listing available depots and their contents

After creating a depot, you can verify that the depot is visible to your clients by executing the **`swlist`** command. Other hosts on the network can use the same command to see which depots are available from your server.

```
# swlist -l depot @ svr
# Initializing...
# tgt "myhost.hp.com" has the following depot(s):
  /mydepot
  /myappdepot
```

You can also list the contents of a specific depot using a variation on the same command. This feature, too, is available to anyone on the network.

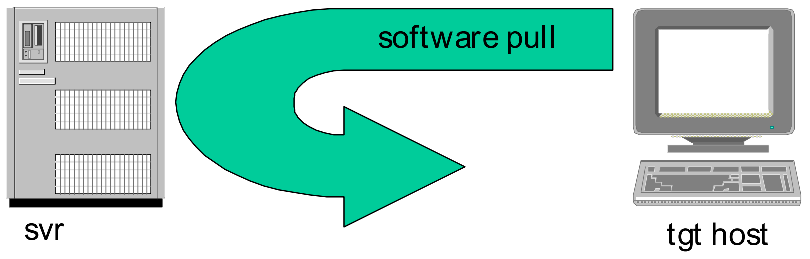
```
# swlist -l product -s svr:/mydepot
# tgt:  hpvc:/mydepot
# Bundle(s):
  100BaseT-00     B.11.11.01     EISA 100BaseT
  100BaseT-01     B.11.11.01     HP-PB 100BaseT
```



Pulling Software from a Depot

Once the depot server has been configured, any host on the network can "pull" software from the depot server via the swinstall command.

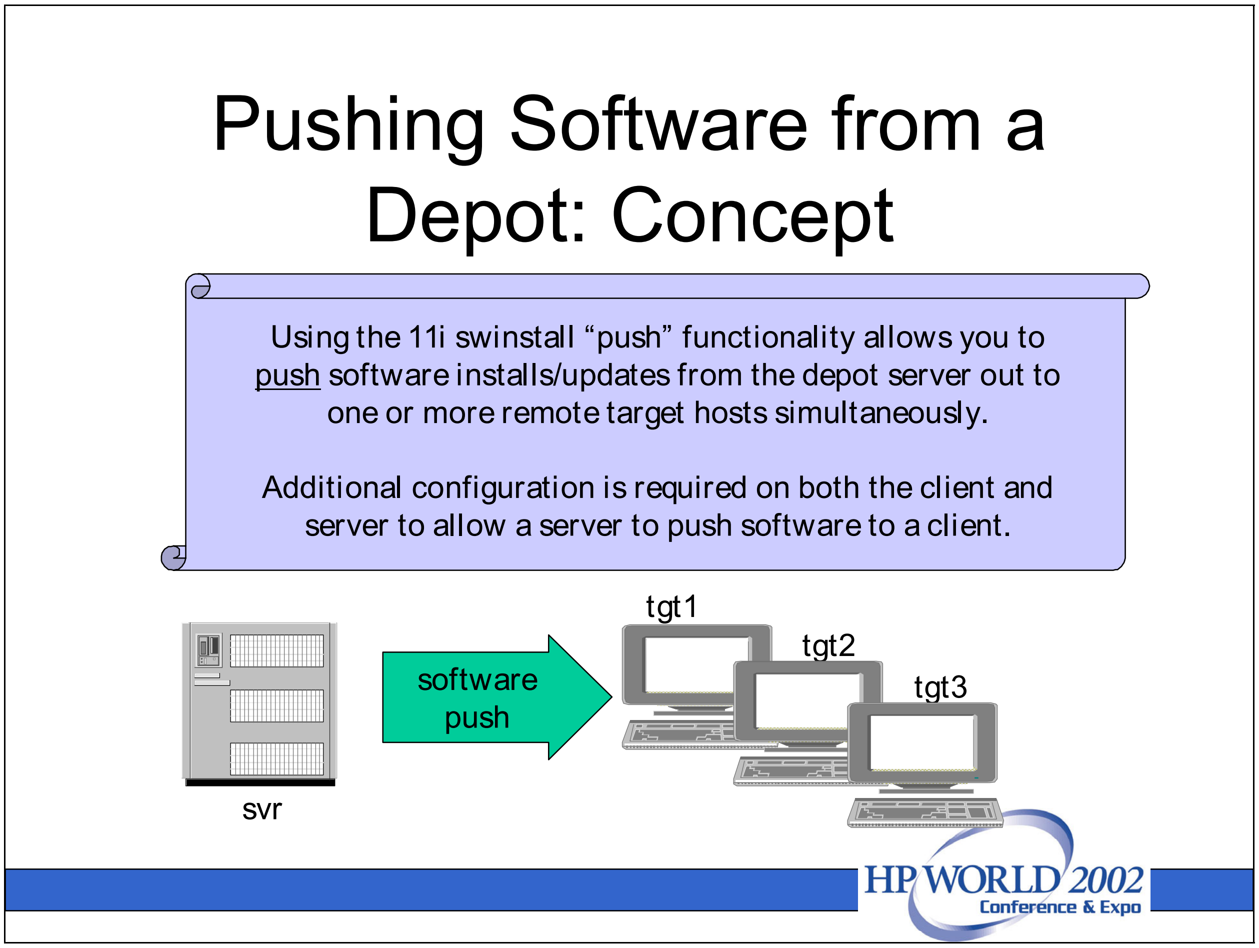
```
tgt# swinstall -s svr:/mydepot
                  -x autoreboot=true FooProd
```



Once you have configured your depot server, your clients can use the **swinstall** command to pull software from your depots, just as you would install software from a CD. Simply specify **server:/depotpath** after the **-s** source option.

```
# swinstall -s svr:/mydepot -x autoreboot=true FooProd
```

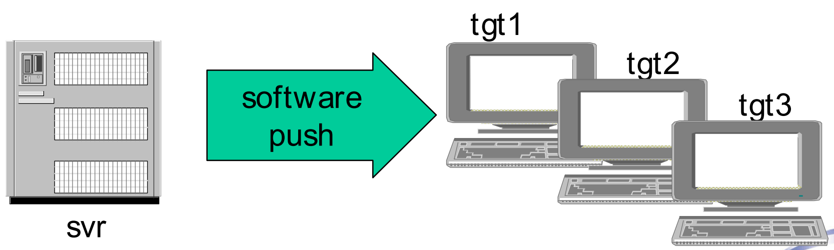
After analyzing the requirements of the selected product(s) and auto-selecting dependencies and patches from the depot, **swinstall** installs and configures the software on your system. If the product or bundle contains a kernel fileset, **swinstall** will automatically reboot your system.



Pushing Software from a Depot: Concept

Using the 11i swinstall "push" functionality allows you to push software installs/updates from the depot server out to one or more remote target hosts simultaneously.

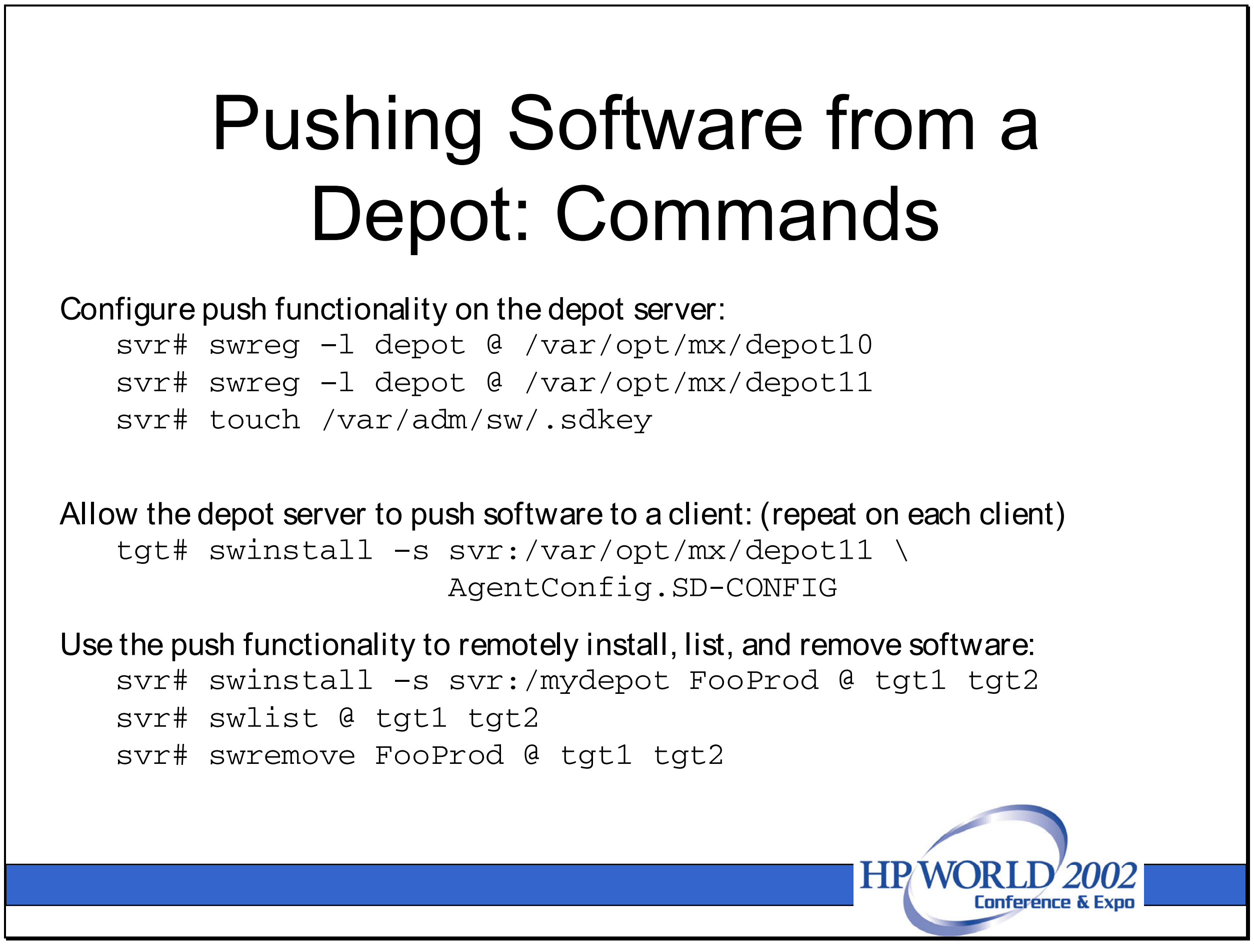
Additional configuration is required on both the client and server to allow a server to push software to a client.



HP WORLD 2002 Conference & Expo

The 11i version of the **`swinstall`** command was enhanced to provide the ability to *push* software to remote systems from a depot server. The **`swremove`**, **`swcopy`**, and **`swlist`** all are capable now of performing remote operations, too, both from the command line and via the interactive GUI interface. You can monitor the results of a remote operation using the swjob command, or the associated Job Browser GUI.

This new functionality allows you to manage software and patches on multiple systems from one central depot server. With sufficient network bandwidth, you could potentially maintain consistent software loads on hundreds of systems scattered across your enterprise from one central depot server!



Pushing Software from a Depot: Commands

Configure push functionality on the depot server:

```
svr# swreg -l depot @ /var/opt/mx/depot10
svr# swreg -l depot @ /var/opt/mx/depot11
svr# touch /var/adm/sw/.sdkey
```

Allow the depot server to push software to a client: (repeat on each client)

```
tgt# swinstall -s svr:/var/opt/mx/depot11 \
                  AgentConfig.SD-CONFIG
```

Use the push functionality to remotely install, list, and remove software:

```
svr# swinstall -s svr:/mydepot FooProd @ tgt1 tgt2
svr# swlist @ tgt1 tgt2
svr# swremove FooProd @ tgt1 tgt2
```

Configuring push functionality on the depot server

Several steps are required to enable the new push functionality on the depot server. First, register the **`depot10`** and **`depot11`** depots on the depot server. These depot directories, which are included in a standard 11i install, contain filesets that must be installed on each client that you intend to push to. **`depot10`** contains software for 10.x clients, and **`depot11`** contains software for 11.x clients. Registering the depots makes them accessible to remote clients. Normally, when you create a new depot with **`swcopy`**, the depot is registered for you automatically.

```
svr# swreg -l depot @/var/opt/mx/depot10
svr# swreg -l depot @ /var/opt/mx/depot11
```

Next, touch a file called **`/var/adm/sw/.sdkey`**. When you run the **`swinstall`** GUI, **`swinstall`** checks to determine if this file exists. If the file does exist, **`swinstall`** launches a somewhat modified GUI that allows you to specify one or more remote target hosts to push software to. Without this file, **`swinstall`** launches the traditional GUI interface and assumes that all selected software should be installed on the localhost. If you don't use the **`swinstall`** GUI, you can skip this step.

```
# touch /var/adm/sw/.sdkey
```

Allow the depot server to push software to a client

The depot server isn't allowed to push software to a target client until the client explicitly allows the depot server to do push installs. This requires several changes to the SD-UX Access Control Lists (unrelated to HFS or JFS ACLs). The SD-UX ACL mechanism is fairly sophisticated, and can't be covered in the short time available in this seminar. For more information, see the *Software Distributor Administrator Guide for HP-UX 11i* (Part Number: B2355-90699) manual on **`http//docs.hp.com`**. Fortunately, SD-UX will set the appropriate ACLs for you if you install the SD-CONFIG fileset from the depot server on the target client.

```
tgt# swinstall -s /var/opt/mx/depot11 AgentConfig.SD-CONFIG
```

Use the push functionality to remotely install, list, and remove software

After you have configured both the depot server and target clients, you can begin pushing software from the depot server. Here are a few examples:

```
# swinstall -s svr:/mydepot FooProd @ tgt1 tgt2
# swlist @ tgt1tgt2
# swremove FooProd @ tgt1 tgt2
```

If you created the **`/var/adm/sw/.sdkey`** file above, then the GUI interface for each of these commands will include a new screen that allows you to select a target host for the SD-UX operations.

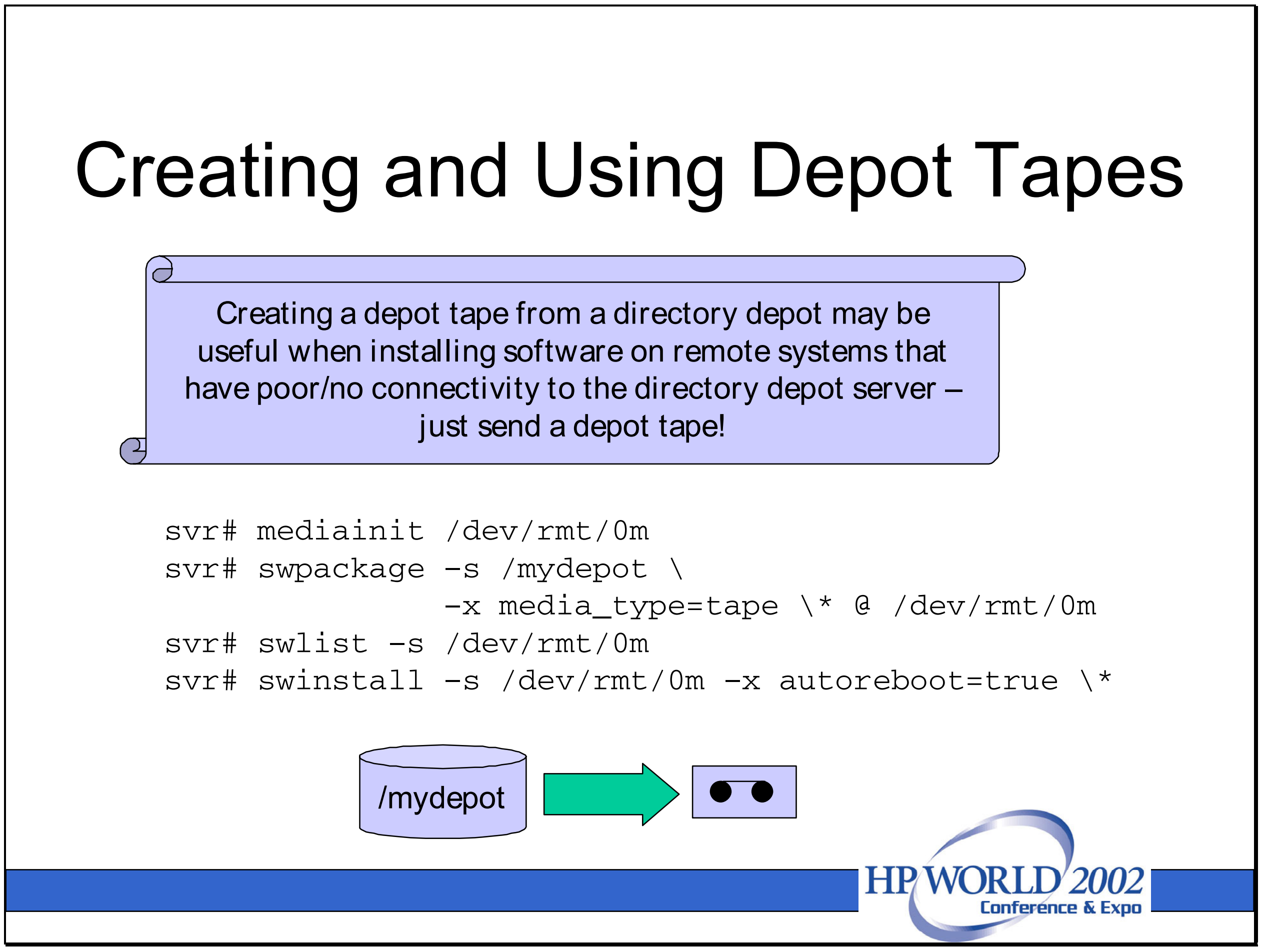
Limitations

- You cannot use remote operations to directly "push" an HP-UX OS update to remote systems.

- The **`swinstall -x patch_match_target`** option works with the push functionality, but you can only push to one remote system at a time.

- The following commands don't support the SD-UX push functionality: **`update-ux, install-sd, swpackage, swmodify`**

- You can only push software from an 11i depot server, though the target hosts can be 10.20, 11.00, or 11i.



Creating and Using Depot Tapes

Creating a depot tape from a directory depot may be useful when installing software on remote systems that have poor/no connectivity to the directory depot server – just send a depot tape!

```
svr# mediainit /dev/rmt/0m
svr# swpackage -s /mydepot \
               -x media_type=tape \* @ /dev/rmt/0m
svr# swlist -s /dev/rmt/0m
svr# swinstall -s /dev/rmt/0m -x autoreboot=true \*
```

Most of the time you will be managing depots for use within your own installation. However there may be times when you want to transport or deliver a series of patches to either one of your own customers or even to send a patch depot to your own colleagues for use on a remote site. One convenient way to accomplish this is to produce a tape in the SD format.

To accomplish this, we use the **`swpackage`** command. **`swpackage`** may be used two ways: (a) to copy software from an existing directory depot to a tape or **`.depot`** file, or (b) to package a newly written application in the SDUX format using a Product Specification File. For more information on PSF files and packaging new software products with **`swpackage`**, see the *Software Distributor Administrator Guide for HP-UX 11i* (Part Number: B2355-90699) available at **`http://docs.hp.com`**. Creating a PSF file for a new software product is beyond the scope of this paper. Here, we will simply examine the process necessary to copy software from a directory depot to tape or a depot file.

Creating a depot tape

The example below only copies FooProd from the **`/mydepot`** depot to a depot tape in **`/dev/rmt/0m`**. If you only want to include all of the products in **`/mydepot`** on your tape depot, replace **`FooProd`** with **`'*'`**.

```
svr# swpackage -s /mydepot -x media_type=tape FooProd @ /dev/rmt/0m
```

Unlike **`swcopy`**, **`swpackage`** WILL NOT resolve any patch dependencies. You must manually list every patch that you wish to include on the depot tape.

Listing the contents of a depot tape

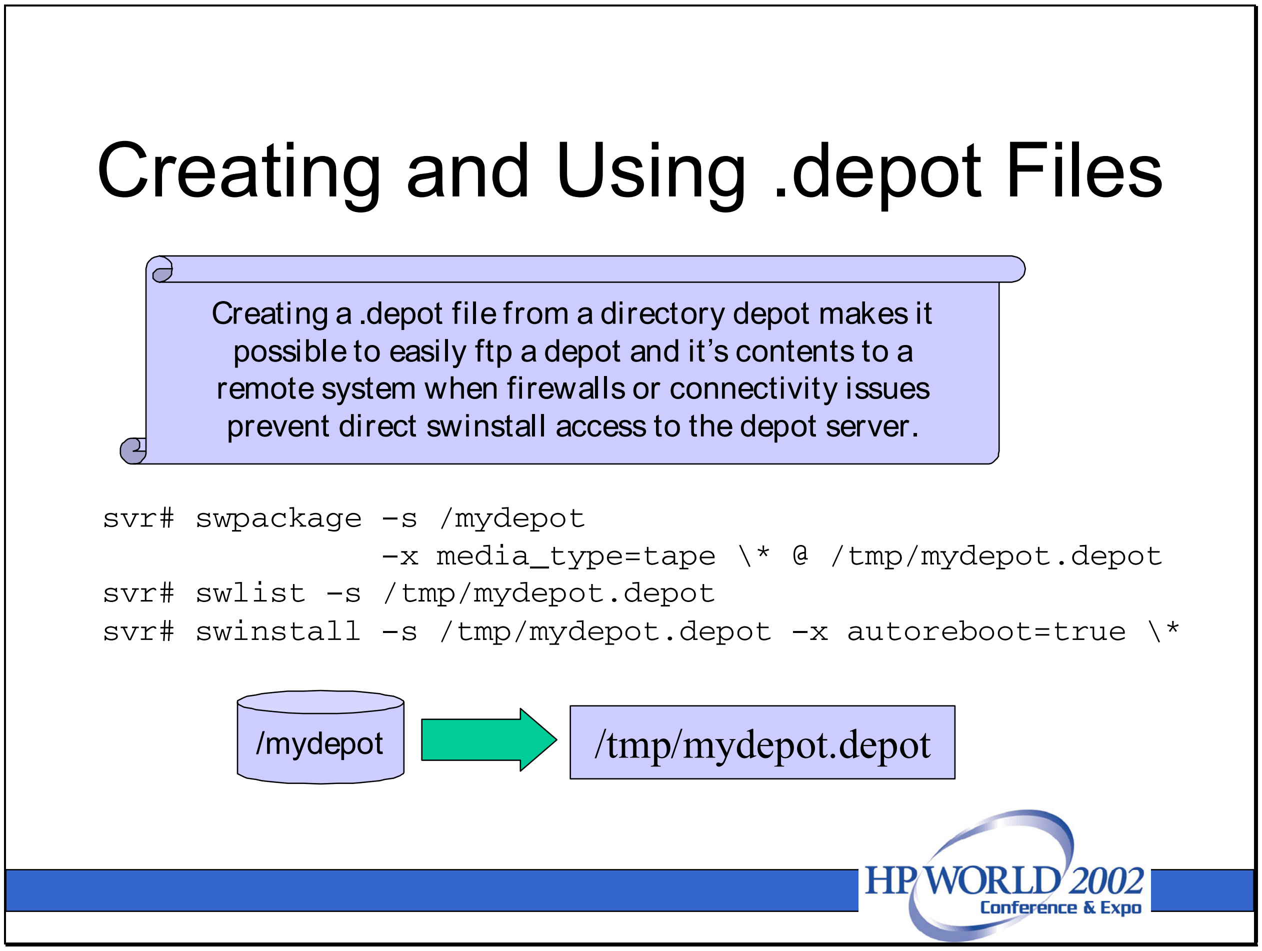
You can list the contents of a depot tape with the **`swlist`** command:

```
svr# swlist –s /dev/rmt/0m
```

Installing software from a depot tape

The **`swinstall`** syntax for installing software from a depot tape is very similar to the syntax used to install from a directory depot:

```
tgt# swinstall –s /dev/rmt/0m –x autoreboot=true FooProd
```



Creating and Using .depot Files

Creating a .depot file from a directory depot makes it possible to easily ftp a depot and it's contents to a remote system when firewalls or connectivity issues prevent direct swinstall access to the depot server.

```
svr# swpackage -s /mydepot
               -x media_type=tape \* @ /tmp/mydepot.depot
svr# swlist -s /tmp/mydepot.depot
svr# swinstall -s /tmp/mydepot.depot -x autoreboot=true \*
```

It is also possible to create a depot formatted file. This is useful if you wish to make the contents of your depot available to a system that can't access your depot server directly because of firewall or connectivity issues. If the remote system has email or FTP access, you can use **`swpackage`** to create a **`.depot`** file, email or FTP it to the remote machine, then do a **`swinstall`** from the file.

Creating a .depot file

The command required to create the **`.depot`** file looks a lot like the **`swpackage`** command we used on the previous page to create a depot tape.

```
# swpackage -s /software -x media_type=tape \
            FooProd @ /tmp/FooProd.depot
```

Listing the contents of a .depot file

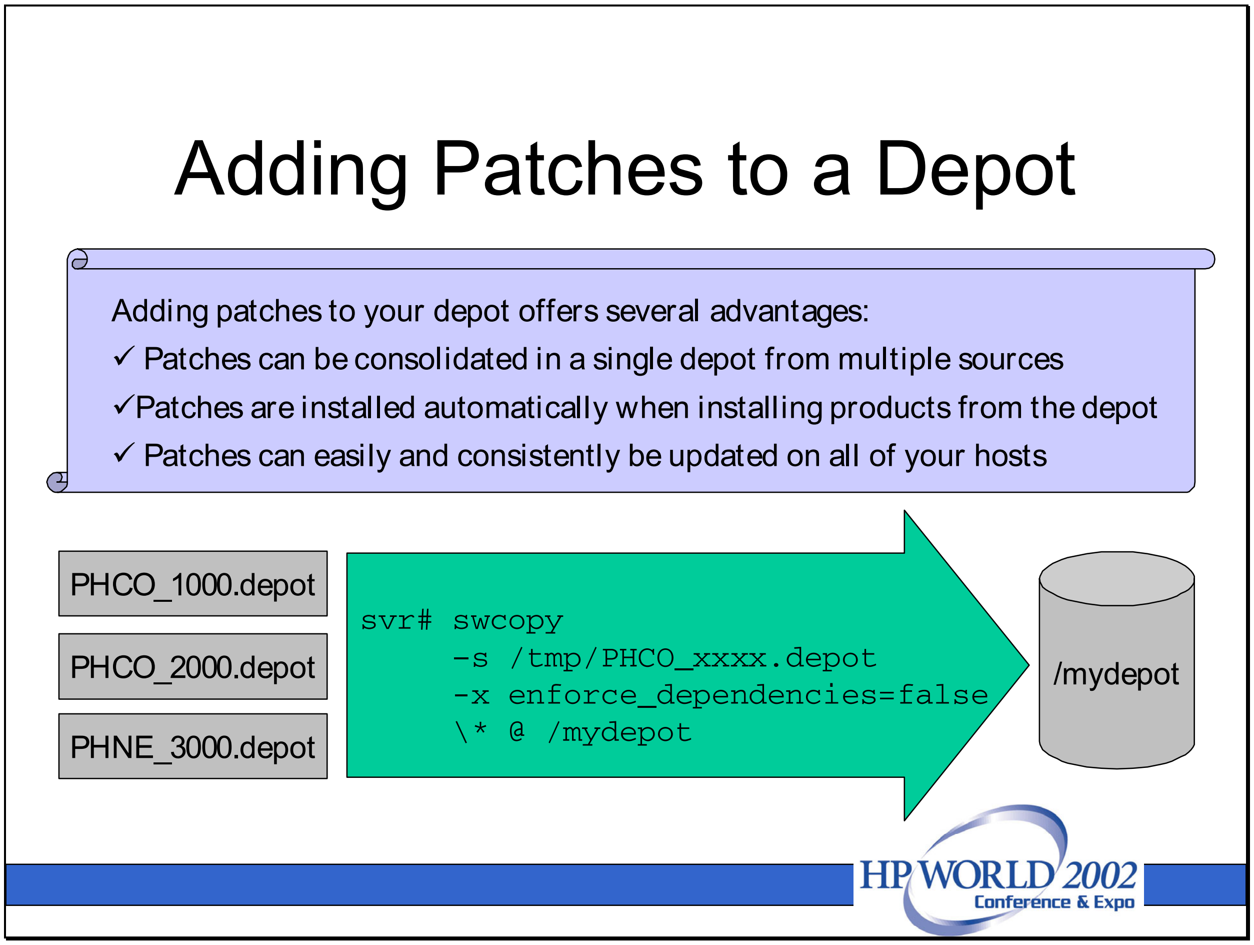
Use the **`swlist`** command to list the contents of a **`.depot`** file.

```
# swlist -s /tmp/FooProd.depot
```

Installing software from a .depot file

Use the **`swinstall`** command to install software from the **`.depot`** file.

```
tgt# swinstall -s /tmp/FooProd.depot -x autoreboot=true FooProd
```



Adding Patches to a Depot

Adding patches to your depot offers several advantages:

✓ Patches can be consolidated in a single depot from multiple sources

✓Patches are installed automatically when installing products from the depot

✓ Patches can easily and consistently be updated on all of your hosts

```
svr# swcopy
     -s /tmp/PHCO_xxxx.depot
     -x enforce_dependencies=false
     \* @ /mydepot
```

Although a product-only depot is useful, providing patches as well as products in your SD-UX depots offers even greater power and flexibility:

- Some of the patches that you use in your shop probably come from the Support+ CD, some may be downloaded from the ITRC, and some may be pulled from patch tapes. Using an SD-UX depot, it becomes possible to consolidate patches from all of those sources into a single network depot.

- When installing a new product from a depot, the default **`autoselect_patches=true`** option on **`swinstall`** automatically selects and installs any matching patches from the depot, too:

```
tgt# swinstall -s svr:/mydepot \
               -x autoselect_patches=true \
               -x autoreboot=true FooProd
```

  Since **`swinstall`** installs both the product and its patches simultaneously, this minimizes the number of **`swinstall`** sessions and reboots necessary to install the product and necessary patches. If the auto-selected patches have dependencies,

swinstall automatically selects the dependents, too.

- After a product has been initially installed, depots simplify patch updates, too. Client administrators can simply use the **swinstall -x patch_match_target** command to automatically select patches from the depot that match products already installed on the target system:

```
tgt# swinstall -s svr:/mydepot \
               -x patch_match_target=true \
               -x autoreboot=true \
```

Adding patches to your depot

Patches may be added to your depots in much the same way that products were added to your depots. The example below copies PHCO_1000 to **/mydepot** from a **.depot** file that was downloaded and unshar'ed from the ITRC:

```
svr# swcopy -s /tmp/PHCO_1000.depot \
            -x enforce_dependencies=false \* @ /mydepot
```

This next example copies all the patches from a Support+ GOLDBASE11i depot to **/mydepot**:

```
svr# swcopy -s /cd/GOLDBASE11i \
            -x enforce_dependencies=false \* @ /mydepot
```

This last example copies all the patches from a patch tape to **/mydepot**:

```
svr# swcopy -s /dev/rmt/0m \
            -x enforce_dependencies=false \* @ /mydepot
```

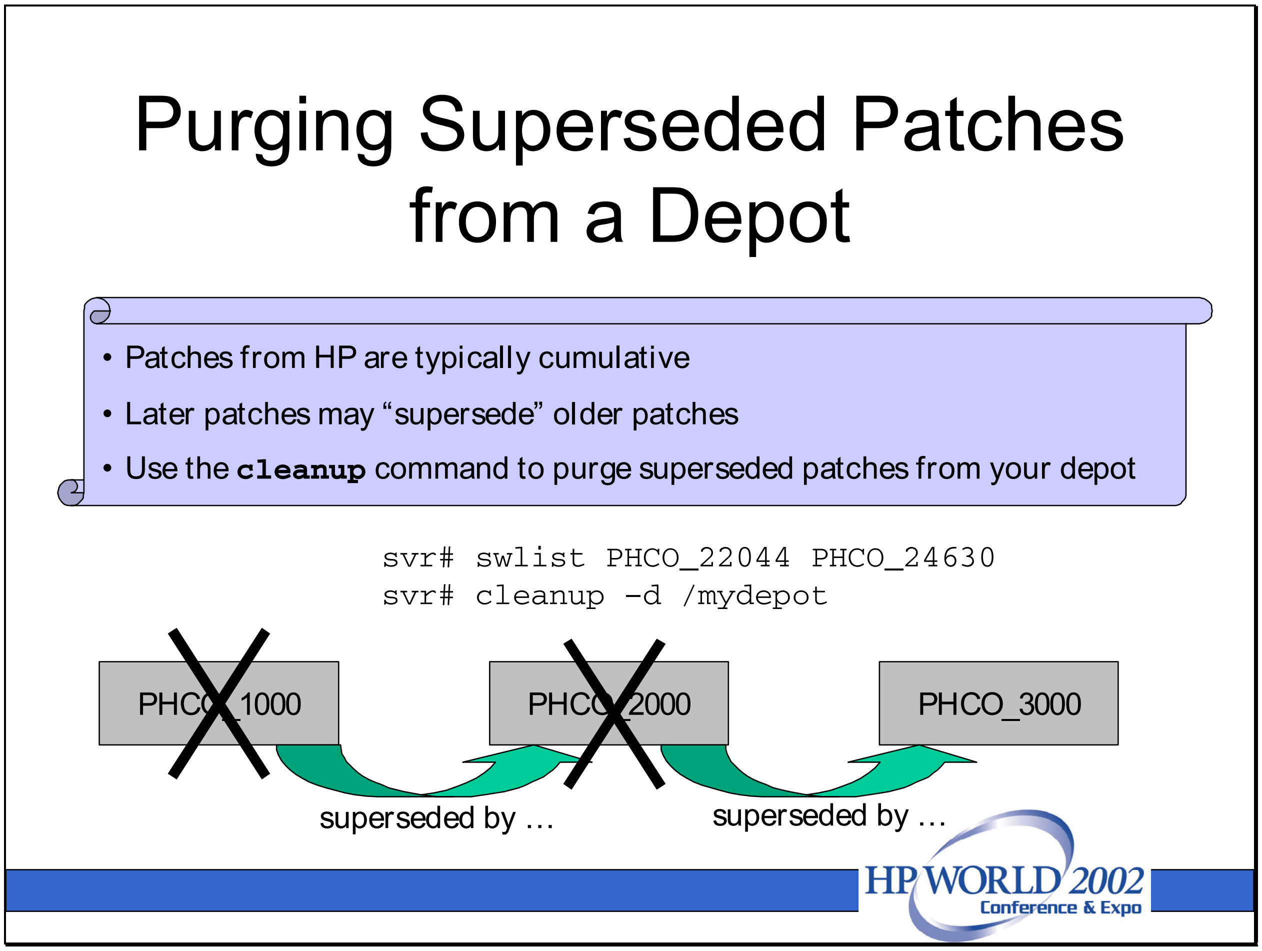
Patch dependencies

Note that all of the **swcopy** examples above included the **-x enforce_dependencies=false** option.

Oftentimes, in order for an HP-UX patch to function properly, one or more additional patches may be necessary to meet the patch's dependencies. These dependencies are typically documented in the patch's **.text** file, and on the ITRC patch database web page. By default, the **swcopy** command won't copy a patch to a depot unless the patch's dependencies can be resolved in the depot.

When setting up a patch depot, it is common to have a collection of downloaded **.depot** files, each of which contains a single patch. In such a situation it will be necessary to override **swcopy** dependency checking to setup the depot.

When clients **swinstall** patches from the depot, however, the **swinstall** command must verify that dependencies have been met. Although it safe and sometimes necessary to override dependency checking on **swcopy**, it is very dangerous to override dependency checking when running **`swinstall`**. Doing so can render a system unstable and liable to many future problems.



Purging Superseded Patches from a Depot

- Patches from HP are typically cumulative
- Later patches may "supersede" older patches
- Use the **`cleanup`** command to purge superseded patches from your depot

```
svr# swlist PHCO_22044 PHCO_24630
svr# cleanup -d /mydepot
```

PHCO_1000
PHCO_2000
PHCO_3000
superseded by …
superseded by …
HP WORLD 2002 Conference & Expo

HP-UX patches are designed to be cumulative. If a new defect is identified in a fileset, and the fileset already has an existing patch, a new patch will be released which incorporates all of the fixes included in the existing patch, plus any changes needed to correct the new defect. Thus, oftentimes a patch will "supersede" one or more older patches. Hosts need only install the most recent patch in the supersession chain.

When multiple patches in a supersession chain are available from a depot, **`swinstall`** will automatically select the most recent patch.

However, in order to conserve disk space on the depot server and expedite client **`swinstall`**s, many administrators choose to regularly purge superseded patches from their depots. The **`cleanup`** command automates this process.

Installing and running cleanup

1. The cleanup command isn't included on the HP-UX install media; it must be downloaded and installed from the ITRC in the form of a patch. Download the appropriate patch from **`http://itrc.hp.com`**:

   11.00 PHCO_22044

11.11   PHCO_24630

2. Unshar and install the patch:

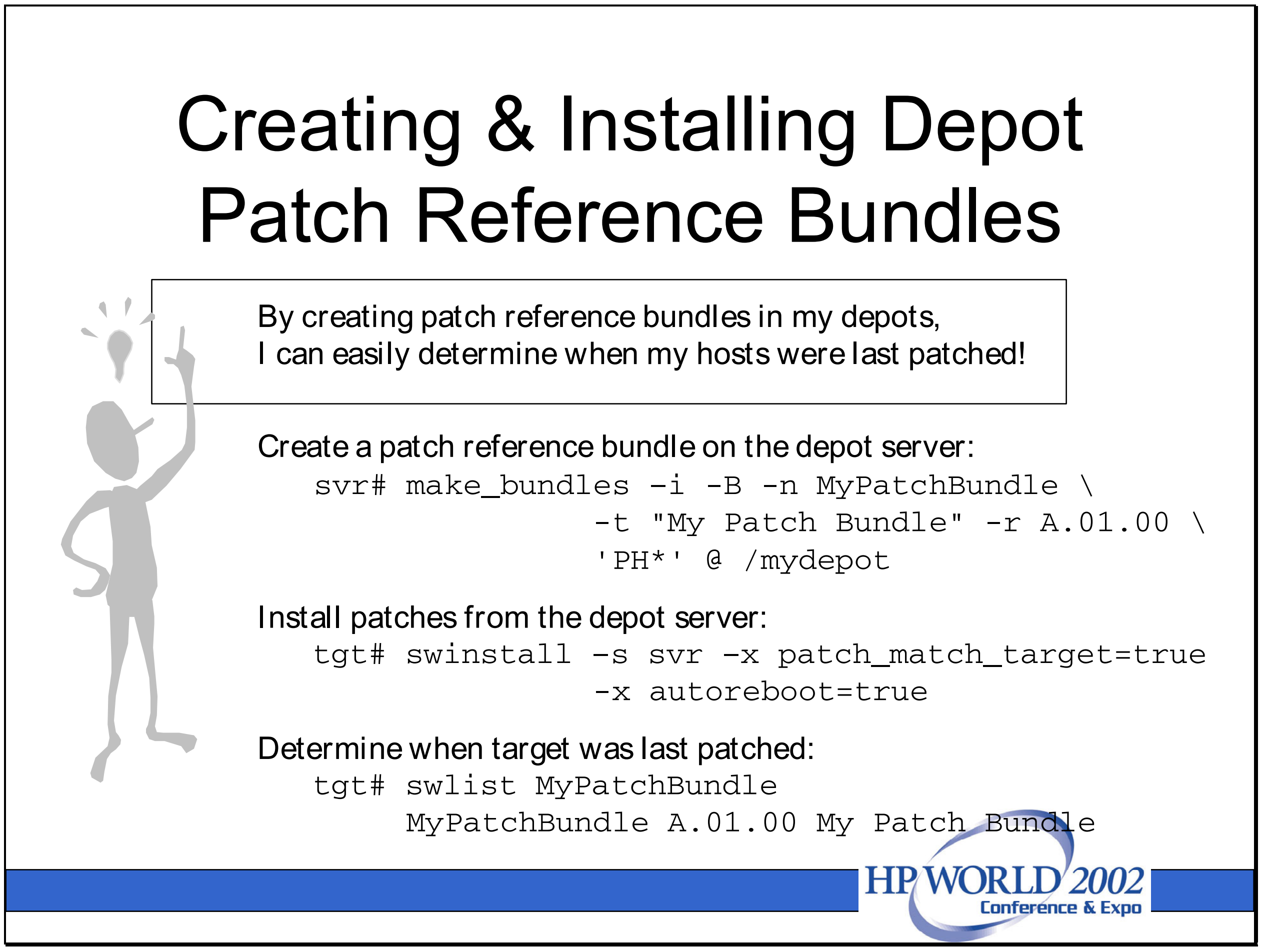
```
# cd /tmp
# sh /tmp/PHCO_24630
# swinstall -s /tmp/PHCO_24630.depot PHCO_24630
```

3. Run the **`cleanup`** command with the **`-d`** option to purge superseded patches from your depot. Note that **`cleanup`** will only remove a patch if the superseding patch is in your depot! If you simply wish to view a list of superseded depot patches without removing them, include the **`-p`** (preview) option, too.

```
# cleanup -d /mydepot
### Cleanup program started at 07/20/02  15:09:27
Cleanup of depot '/mydepot'.
Obtaining the list of patches in the depot: /mydepot ...done.
Obtaining the list of superseded 11.X patches in the depot:
     /mydepot
...The following superseded patches exist in the depot:
====================================================
PHCO_1000 superseded by PHCO_2000
PHCO_2000 superseded by PHCO_3000

Please be patient; this may take several minutes.

Removing superseded 11.X patches from depot: /mydepot ...done.
The superseded 11.X patches have been removed from the depot:
     /mydepot.
All information has been logged to /var/adm/cleanup.log.
### Cleanup program completed at 07/20/02  15:09:27
```



Creating & Installing Depot Patch Reference Bundles

By creating patch reference bundles in my depots,
I can easily determine when my hosts were last patched!

Create a patch reference bundle on the depot server:

```
   svr# make_bundles -i -B -n MyPatchBundle \
                  -t "My Patch Bundle" -r A.01.00 \
                  'PH*' @ /mydepot
```

Install patches from the depot server:

```
   tgt# swinstall -s svr -x patch_match_target=true
                  -x autoreboot=true
```

Determine when target was last patched:

```
   tgt# swlist MyPatchBundle
        MyPatchBundle A.01.00 My Patch Bundle
```

Over time, you will probably find yourself continuously adding new patches to your SD-UX depots. In order to more easily track which patches were installed on which systems, some administrators bundle patches together to form patch bundles in their SD-UX depots. This makes it very easy to determine whether a system on the network has the latest patches installed. Simply compare the patch bundle revision number on the host in question to the patch bundle revision number on the depot server!

Creating a Custom Patch Bundle

First, verify that you have the Ignite-UX product installed on your system. The standard SD-UX tools aren't able to create bundles, but the **`make_bundles`** command in Ignite-UX can.

```
svr# swlist Ignite-UX-11-11
```

Next, use the **`swcopy`** command to copy the desired patches into your depot. Remove superseded and recalled patches if necessary.

```
svr# swcopy -s /tmp/PHCO_1000.depot PHCO_1000 @ /mydepot
```

Now run the **`make_bundles`** command to create the patch bundle.

```
svr# make_bundles -i \
                  -B \
                  -n MyPatchBundle \
                  -t "My Patch Bundle" \
                  -r A.01.00 \
                  'PH*' @ /mydepot
```

Here's an explanation of the **`make_bundles`** options that are included in the command above:

`-i`	Causes **`make_bundles`** to set the **`is_reference`** attribute to true. Bundle wrappers with **`is_reference`** set to true will be automatically installed anytime any fileset or product within the bundle is installed. If **`is_reference`** is false then only the filesets and products selected will be installed, and the bundle wrapper will not be installed unless all the filesets within the bundle are installed.
`-B`	Causes **`make_bundles`** to create a bundle containing all products/filesets in the depot or those specified on the command line, even if some of the specified products are already included in other bundles.
`-n` ***`name`***	Specifies the name of the bundle, which may be up to 16 characters.
`-t` ***`title`***	Specifies a one-line title or description of the bundle.
`-r` ***`revision`***	Specifies a revision number for the bundle. Revision numbers may include letters, numbers, and dots.
`'PH*'`	Specifies which products should be included in the depot. Using the **`PH*`** ensures that all patches in the depot are included in the bundle. Alternatively, you may provide an explicit list of patches to include in the bundle.
`@ /mydepot`	Specifies the pathname of the depot containing the patches to be bundled.

Updating a Patch Bundle

If you need to add additional patches to your path bundle, simply **`swcopy`** them to the depot, then re-run **`make_bundles`**. Be sure to increment the revision number!

```
svr# swcopy -s /tmp/PHCO_2000.depot PHCO_2000 @ /mydepot
```

Now run the **`make_bundles`** command to create the patch bundle.

```
# make_bundles -i \
               -B \
               -n MyPatchBundle \
               -t "My Patch Bundle" \
               -r A.02.00 \
               'PH*' @ /mydepot
```

Installing Patch Bundles

You can install patch bundles just as you would any other patches:

```
tgt# swinstall -s /software \
               -x patch_match_target=true \
               -x autoreboot=true
```

Note that we didn't have to explicitly select the patch bundle name. **`-x patch_match_target=true`** will automatically select all patches from the depot that match software currently installed on the target. Since the **`is_reference=true`** bundle attribute was set, the patch bundle wrapper will automatically be installed on the target if *any* of the patches in that bundle are installed.

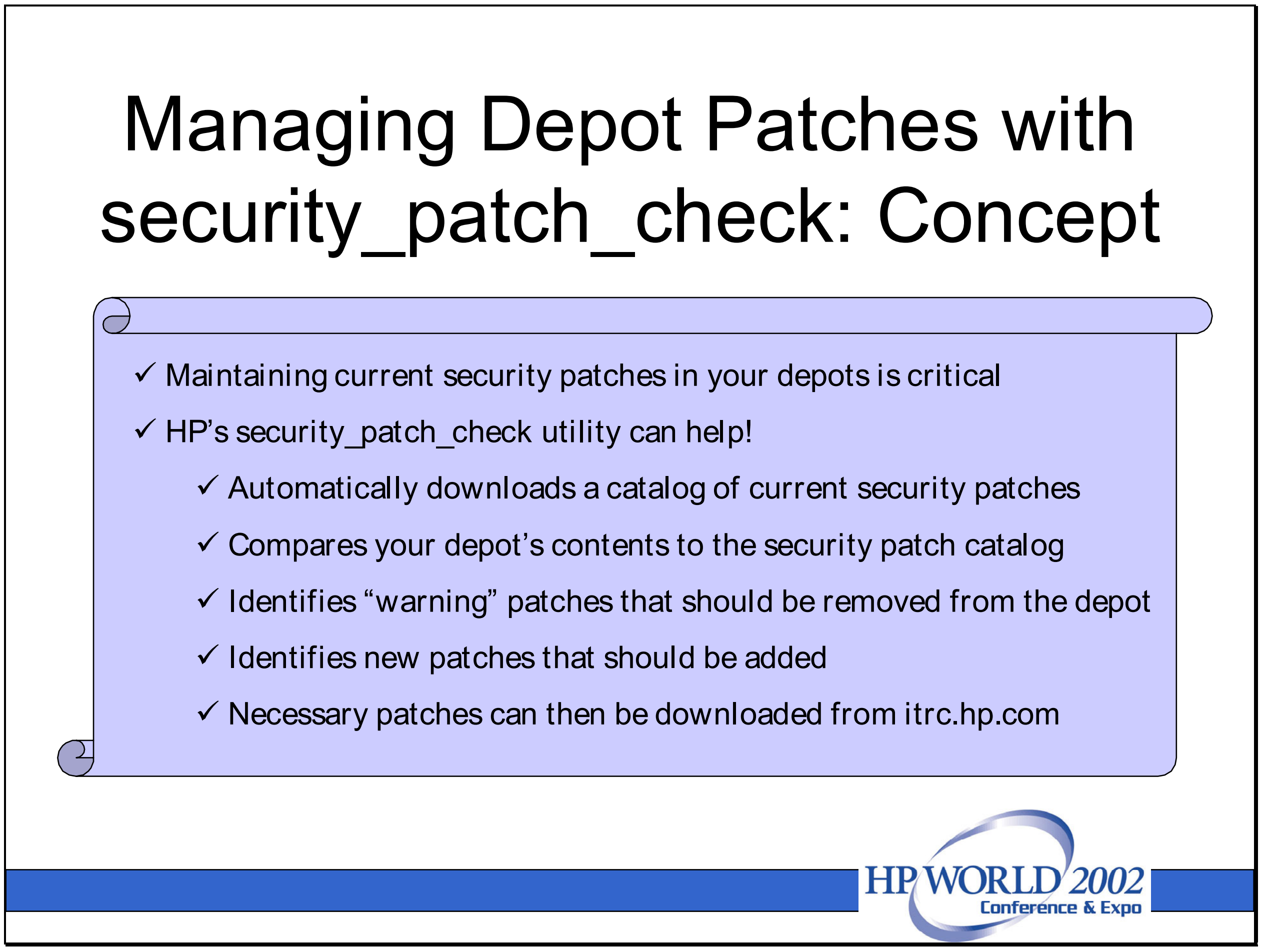
Viewing the Current Patch Bundle Level

You can determine which patch reference bundle is installed on your system via the **`swlist`** command.

```
# swlist MyPatchBundle
# Initializing...
# Contacting target "r812c20"...
#
# Target:  r812c20:/tmp/depot
#
#
# Bundle(s):
#

  MyPatchBundle          A.01.00        My Patch Bundle
```

Note that the existence of the patch reference bundle doesn't necessarily mean that all of the patches that were originally included in the bundle were actually installed on your target. **`swinstall`** only installs the patches that match software installed on the target. However, if the revision number on the depot server is greater than the revision number on a target client, then the client may benefit from a patch update.



Managing Depot Patches with security_patch_check: Concept

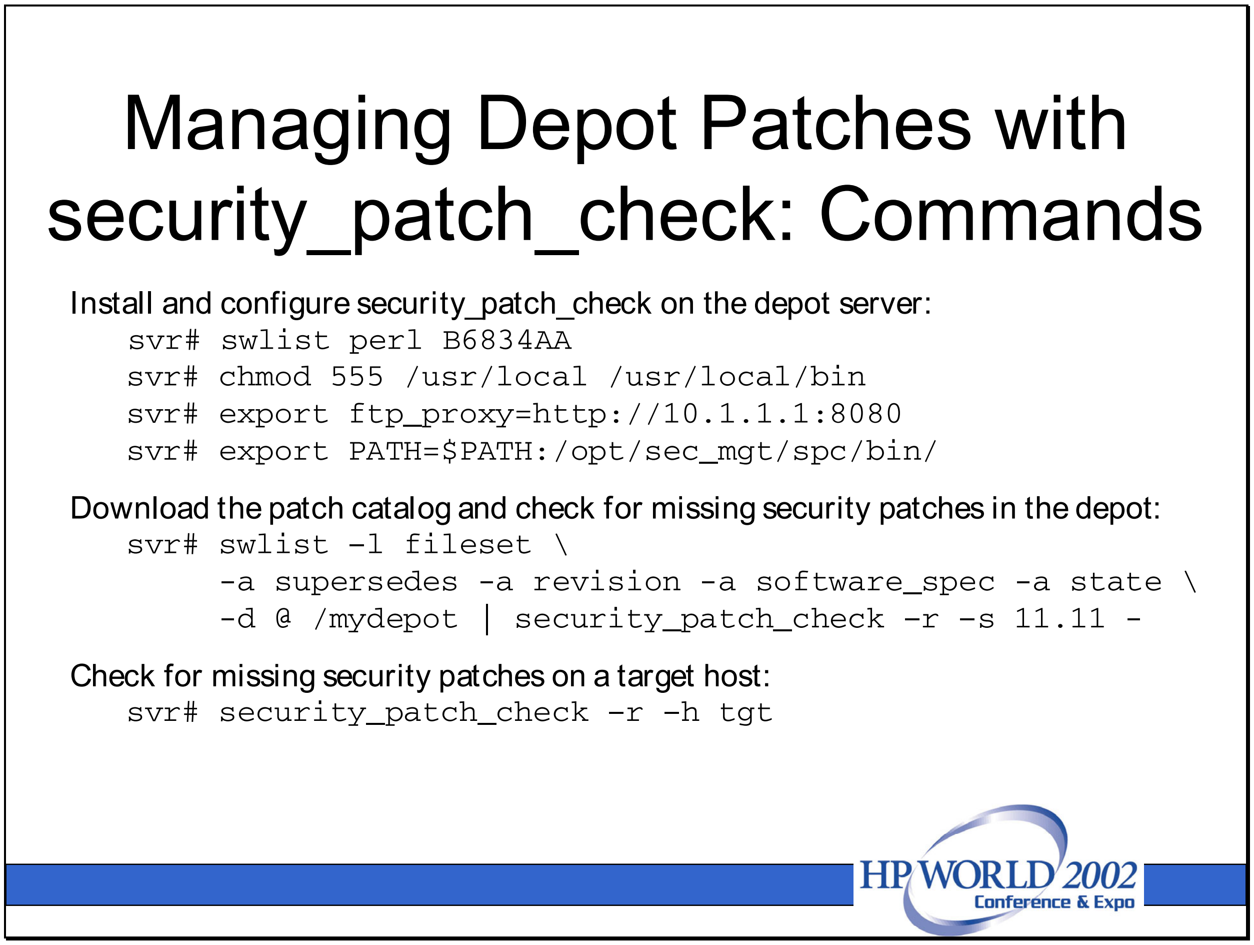
- ✓ Maintaining current security patches in your depots is critical
- ✓ HP's security_patch_check utility can help!
  - ✓ Automatically downloads a catalog of current security patches
  - ✓ Compares your depot's contents to the security patch catalog
  - ✓ Identifies "warning" patches that should be removed from the depot
  - ✓ Identifies new patches that should be added
  - ✓ Necessary patches can then be downloaded from itrc.hp.com

Security patches are among the most important patches that you need to maintain in your depots; a missing security patch could jeopardize the integrity of all the systems on your network! Managing security patches, though, can be a serious challenge since new security patches are oftentimes released on a weekly, if not daily, basis. **`security_patch_check`** was specifically designed by HP to address this challenge.

`security_patch_check` is a Perl script that runs on HP-UX 11.x systems. It automatically downloads a list of current security patches from the ITRC website, then compares the list against either the contents of an SD-UX depot on a depot server, or the installed patches listed in the installed product database. It then generates a report listing (a) the missing security patches that apply to products on your system or depot, and (b) patches with security warnings that should be removed from your host or depot. **`security_patch_check`** doesn't actually add or remove any patches for you automatically; it simply generates a report. You are then responsible for downloading and installing/removing the appropriate patches manually using **`swinstall`** and/or **`swremove`**.

Limitations

Installing the patches that **`security_patch_check`** recommends addresses only those vulnerabilities that are closed by patches. The security bulletins and advisories from HP sometimes contain other actions (manual steps) to close vulnerabilities. Thus, each advisory from the archive of previously-released security advisories must be examined to determine if any manual steps are required. This archive can be obtained from **`http://itrc.hp.com/cki/bin/doc.pl/screen=ckiSecurityBulletin`**.



Managing Depot Patches with security_patch_check: Commands

Install and configure security_patch_check on the depot server:

```
svr# swlist perl B6834AA
svr# chmod 555 /usr/local /usr/local/bin
svr# export ftp_proxy=http://10.1.1.1:8080
svr# export PATH=$PATH:/opt/sec_mgt/spc/bin/
```

Download the patch catalog and check for missing security patches in the depot:

```
svr# swlist -l fileset \
     -a supersedes -a revision -a software_spec -a state \
     -d @ /mydepot | security_patch_check -r -s 11.11 -
```

Check for missing security patches on a target host:

```
svr# security_patch_check -r -h tgt
```

Several steps are required to configure and use **`security_patch_check`**.

Install and configure `security_patch_check` on the depot server

`security_patch_check` (bundle B6834AA) can be downloaded and installed from the **`http://software.hp.com`** website. You will also need to install Perl version 5.005 or greater, which is also available from **`http://software.hp.com`**. The version of Perl on the HP site includes the Perl LWP module, which allows **`security_patch_check`** to download the security patch catalog from the ITRC via an FTP proxy. If **`security_patch_check`** generates errors at runtime, try downloading perl from the HP site to ensure you have the right version, with the required modules.

```
# swlist B6834AA perl
```

Next, verify that the permissions on **`/usr/local`** and **`/usr/local/bin`** are 555 (by default, these directories have 777 permissions in HP-UX). Since perl is installed in **`/usr/local`** and **`/usr/local/bin`**, it is important that these directories be properly secured.

```
# chmod 555 /usr/local /usr/local/bin
```

If you don't have a direct connection to the Internet, it may be necessary to set the **ftp_proxy** variable equal to the IP address and port number of your FTP proxy server. You may be able to determine your proxy server's address by checking your browser's preference screen. If you plan to run **security_patch_check** on a regular basis, you may want to define this variable in your **~/.profile** file. Note that a web proxy generally uses the http protocol (even for proxying ftp data).

```
# export ftp_proxy=http://10.1.1.1:8080
```

While editing your **~/.profile**, add the **/opt/sec_mgt/spc/bin** directory to your PATH variable.

```
# export PATH=$PATH:/opt/sec_mgt/spc/bin
```

Download the security patch catalog and check for missing patches in the depot

The **security_patch_check** command was originally designed to check the patches installed on a host, rather than the patches available in a depot. Since we are attempting to check security patches in a depot, we must manually list the products and patches in the depot with **swlist**, and pipe the output to **security_patch_check** as shown below.

```
svr# swlist -l fileset -a supersedes -a revision \
            -a software_spec -a state \
            -d @ /mydepot | security_patch_check -r -s 11.11 -
```

Here's a description of the **security_patch_check** options used in this command:

-r Retrieve a new copy of the security patch catalog. If you leave off this option, **security_patch_check** looks for a **security_catalog** file in your present working directory.

-s Identifies the HP-UX version included in the software depot.

- Specifies that **security_patch_check** should read stdin for a list of depot patches and products to check. Without this option, **security_patch_check** analyzes the installed product database on the localhost.

This command assumes that **security_patch_check** is able to FTP the security patch catalog from the ITRC directly, or via the proxy server defined in **ftp_proxy**. If your host doesn't have FTP access to the ITRC, you will have to manually obtain a copy of the catalog from **ftp://ftp.itrc.hp.com/export/patches/security_catalog**, unzip it, move it to your present working directory, and run **security_patch_check** without the **-r** (retrieve) option. The catalog is currently ~500k in size, and is updated nightly. To ensure that you have all the most current security patches, always download a fresh copy of the catalog when you run **security_patch_check**.

```
svr# netscape ftp://ftp.itrc.hp.com/export/patches/security_catalog
svr# gzip -d security_catalog
```

```
svr# swlist -l fileset -a supersedes -a revision \
            -a software_spec -a state \
            -d @ /mydepot | security_patch_check -s 11.11 -
```

Check security patches on a target host

After verifying that the security patches on your depot are current, you may want to check the patches on other hosts on your network. You can scan the target hosts remotely from the depot server, or locally on the individual targets.

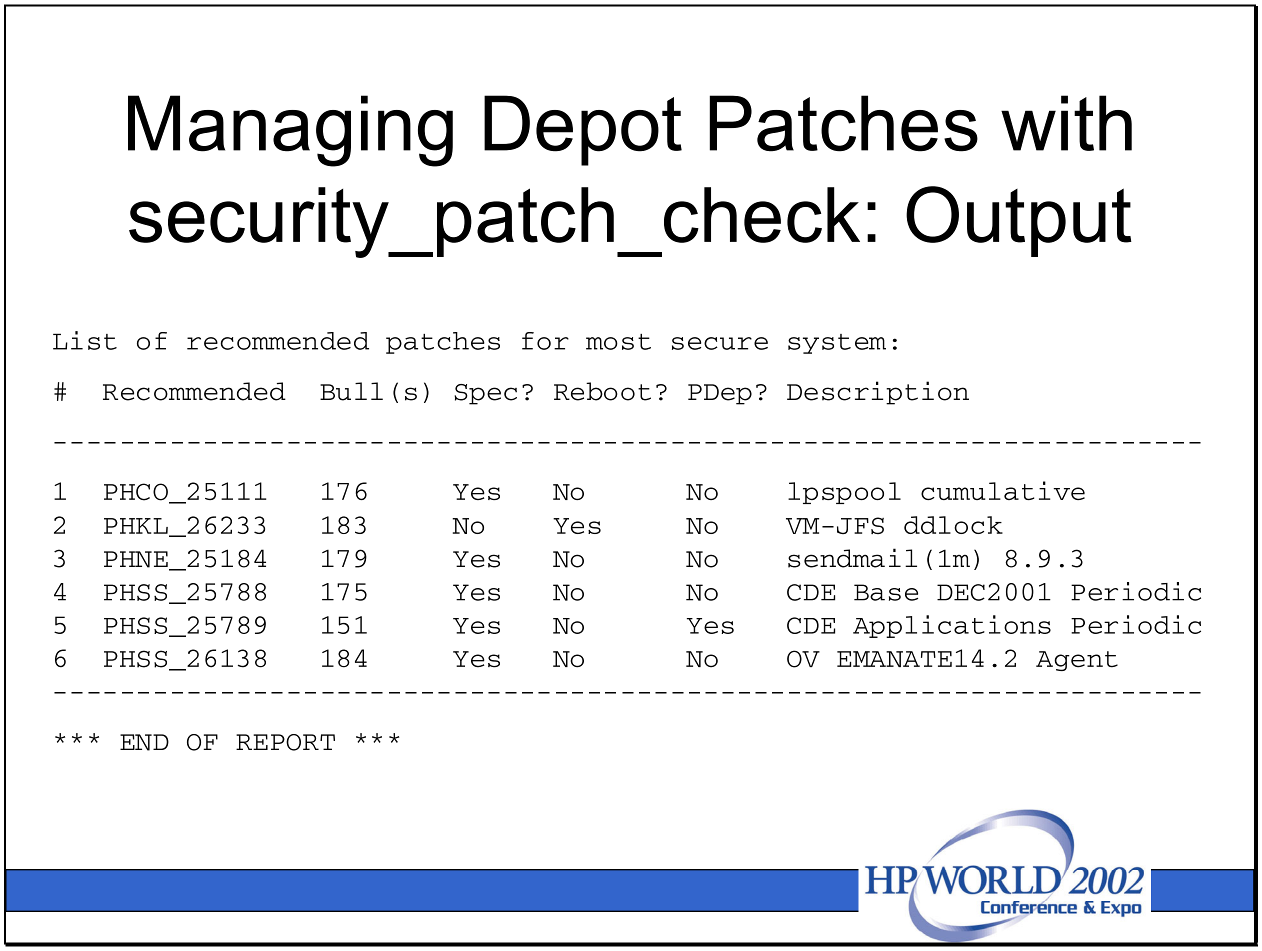
To check a remote target system, use the following (leave off **`-r`** if you already have a copy of **`security_catalog`** in your present working directory):

```
svr# security_patch_check -r -h tgt1
```

To check the patches on localhost, use the following (leave off **`-r`** if you already have a copy of **`security_catalog`** in your present working directory):

```
tgt# security_patch_check -r
```

You may wish to schedule this command to run as a nightly **`cron`** job, emailing the results to root, to ensure that your systems always have the latest security patches.



Managing Depot Patches with security_patch_check: Output

```
List of recommended patches for most secure system:

#   Recommended  Bull(s) Spec? Reboot? PDep? Description
----------------------------------------------------------------------
1   PHCO_25111   176     Yes   No      No    lpspool cumulative
2   PHKL_26233   183     No    Yes     No    VM-JFS ddlock
3   PHNE_25184   179     Yes   No      No    sendmail(1m) 8.9.3
4   PHSS_25788   175     Yes   No      No    CDE Base DEC2001 Periodic
5   PHSS_25789   151     Yes   No      Yes   CDE Applications Periodic
6   PHSS_26138   184     Yes   No      No    OV EMANATE14.2 Agent
----------------------------------------------------------------------

*** END OF REPORT ***
```

HP WORLD 2002 Conference & Expo

`security_patch_check` generates a succinct, text-based report identifying possible patch security issues on your system.

The report begins (not shown on slide) with a list of patches in your depot or on your system that have been recalled. Check the ITRC for more information, and consider removing the offending patches.

After listing recalled patches, **`security_patch_check`** displays a table similar to the one shown on the slide listing recommended security patches based on the filesets in your depot or installed on your system. A summary of the fields in the report is given below:

Recommended:	PatchIDs of the patches you should consider adding to your depot/system.
Bull	Security bulletin that describes the problem addressed by the patch. Patch bulletins are available on the ITRC website at **`http://itrc.hp.com/cki/bin/doc.pl/screen=ckiSecurityBulletin`**.
Spec	Does the patch have special installation instructions?
Reboot	Does the patch require a reboot?

Pdep	Does the patch have any patch dependencies?
Description	What problem does the patch correct?

After reading the report, you can download the appropriate patches from the ITRC web patch database.

Following the recommendations of **`security_patch_check`** will result in a system that is up-to-date with HP's security patches. However, the system will not be fully compliant with HP's security bulletins and advisories, and there are additional steps that must be taken to secure the system.

A complete **`security_patch_check`** report is included below:

```
NOTE: Downloading from
      ftp://ftp.itrc.hp.com/export/patches/security_catalog.sync.

NOTE: ftp://ftp.itrc.hp.com/export/patches/security_catalog.sync downloaded to
      ./security_catalog.sync successfully.

NOTE: Downloading from
      ftp://ftp.itrc.hp.com/export/patches/security_catalog.gz.

NOTE: ftp://ftp.itrc.hp.com/export/patches/security_catalog.gz downloaded to
      ./security_catalog.gz successfully.


*** BEGINNING OF SECURITY PATCH CHECK REPORT ***
Report generated by: /opt/sec_mgmt/spc/bin/security_patch_check.pl, run as
root
Analyzed localhost (HP-UX 11.11) from r811c17
Security catalog: ./security_catalog
Security catalog created on: Thu Jul 18 22:13:08 2002
Time of analysis: Sat Jul 20 13:41:24 2002

List of recommended patches for most secure system:
#  Recommended  Bull(s) Spec? Reboot? PDep? Description
-------------------------------------------------------------------------------
1  PHCO_25111   176     Yes   No      No    lpspool subsystem cumulative
2  PHKL_26233   183     No    Yes     No    VM-JFS ddlock, mmap, user limits
3  PHNE_25184   179     Yes   No      No    sendmail(1m) 8.9.3
4  PHSS_25788   175     Yes   No      No    CDE Base DEC2001 Periodic
5  PHSS_25789   151     Yes   No      Yes   CDE Applications DEC2001 Periodic
6  PHSS_26138   184     Yes   No      No    OV EMANATE14.2 Agent Consolidated
7  PHSS_27182   184     Yes   No      Yes   OV EMANATE14.2 snmpdm
8  PHSS_27258   196     Yes   No      Yes   HP DCE/9000 1.8 DCE Client IPv6
-------------------------------------------------------------------------------
*** END OF REPORT ***
NOTE: Security bulletins can be found ordered by number at
      http://itrc.hp.com/cki/bin/doc.pl/screen=ckiSecurityBulletin
```

Questions?

HP WORLD 2002
Conference & Expo