|  |  |  |  |
| --- | --- | --- | --- |
|  | FS – CIO  RELEASE NOTICE  (6610-7) | |  |
| Month XX, XXXX | **CIO – ISO** | | RN-XXXX-XX |
| **Subject:** | | IBM Tivoli Configuration Manager v4.3.1 Install | |
| **Purpose:** | | Create new USDA Forest Service Tivoli Management Region (TMR) server running IBM Tivoli Configuration Manager v4.3.1. | |
| **Support Contact:** | | Customer Help Desk (1-866-945-1354) | |
| **Application Sponsor:** | |  | |
| **OS Level:** | | AIX v5.3 | |
| **Mandatory:** | | No | |
| **Required System Reboot:** | | No | |
| **Service Interruption:** | | No | |
| **Affected Server Types:** | | Tivoli Management Region servers | |
| **Server Team Implementing Update:** | | ESM | |
| **Expected Completion Date:** | | 2010-12-31 | |
| **Environment:** | | This release will be installed in Prod. | |
| **SR #:** | |  | |
| **CR #:** | |  | |
| **If there are any problems installing this release, please contact the following people:** | | Developer’s name: Ron Compos ([rcompos@us.ibm.com](mailto:rcompos@us.ibm.com)) 303-748-4873  Installation tester’s name: | |
| General Overview | | | |
| **Abstract**  USDA Forest Service CM431 TMR Install is a process to create a USDA Forest Service (FS) Configuration Manager v4.3.1 Tivoli Management Region (TMR).  This process may be used to create a hub or spoke TMR server, Gateway managed nodes and endpoints. If a hub (Enterprise) TMR is being built and no managed nodes or endpoints are desired, follow instructions specific to hub servers.  USDA Forest Service CM431 TMR Install requires a TMR server running AIX v5.3 a collection of Gateway Managed Node servers running AIX v5, Red Hat Enterprise Linux v5 (RHEL5) or SUSE Linux Enterprise Server (SLES 10) operating system and endpoint clients running Windows, AIX or Linux. root access is required on all servers. The creation of managed nodes now supports SSH and no longer requires the exec service to be enabled on the target server. Endpoints are installed via SSH as well.  The Tivoli Inventory data will be stored on a Oracle database management server. This necessitates an Oracle 10g database client on the TMR server. All Inventory objects will be created on the TMR server. This differs from previous USDA FS installations which used a dedicated Inventory managed node server.  This release provides a new endpoint version 43102, from the Tivoli Framework Patch 4.3.1-LCF-0002.  **Products/Patches**  The following are comprehensive lists of products and patches installed in the Tivoli Management Region. The product and patch configuration files tmr.cfg and gw.cfg are located in /usr/local/Tivoli/etc/inst. These config files define the specific configuration for TMR server or gateway managed node.  **TMR Products/Patches** (tmr-fresh.cfg)  Products:  Tivoli Management Framework 4.3.1  Inventory Gateway, Version 4.3.1  Inventory, Version 4.3.1  Tivoli Java Client Framework 4.3.1  Java 1.4.2 for Tivoli  Tivoli Java RDBMS Interface Module (JRIM) 4.3.1  JavaHelp 1.0 for Tivoli 4.1  Pristine Manager, Version 4.3.1  Pristine Manager Gateway, Version 4.3.1  Tivoli Management Framework SSLA Version 1.3  Resource Manager, Version 4.3.1  Resource Manager Gateway, Version 4.3.1  Activity Planner, Version 4.3.1  Change Manager, Version 4.3.1  Directory Query, Version 4.3.1  Distribution Status Console, Version 4.3.1  Patch Management, Version 4.3.1  Software Distribution, Version 4.3.1, Build 20081113  Software Distribution Gateway, Version 4.3.1  Software Distribution Software Package Editor, Version 4.3.1  Web Interface, Version 4.3.1  Patches:  Software Distribution Software Package Editor V. 4.3.1,Fix Pack 4.3.1-TIV-SWDJPS-FP0001(U829862 - 2009/10)  Tivoli Framework Patch 4.3.1-LCF-0002 (build 09/23)  Activity Planner, Version 4.3.1, Fix Pack 4.3.1-TIV-APM-FP0001 (U829862 - 2009/10)  Scalable Collection Service, Version 4.3.1, Fix Pack 4.3.1-TIV-CLL-FP0001 (U829862 - 2009/10)  Inventory, Version 4.3.1, Fix Pack 4.3.1-TIV-INV-FP0001 (U829862 - 2009/10)  Inventory Gateway, Version 4.3.1, Fix Pack 4.3.1-TIV-INVGW-FP0001 (U829862 - 2009/10)  Software Distribution Gateway, Version 4.3.1, Fix Pack 4.3.1-TIV-SWDGW-FP0001 (U829862 - 2009/10)  Software Distribution, Version 4.3.1, Fix Pack 4.3.1-TIV-SWDSRV-FP0001 (U829862 - 2009/10)  Tivoli Framework Patch 4.3.1-TMF-0005 (build 11/23)  Tivoli Management Framework SSLA Version 1.3  Scalable Collection Service, Version 4.3.1  **GW Products/Patches** (gw.cfg)  Products:  Tivoli Management Framework 4.3.1  Inventory Gateway, Version 4.3.1  Tivoli Java Client Framework 4.3.1  ava 1.4.2 for Tivoli  Tivoli Java RDBMS Interface Module (JRIM) 4.3.1  JavaHelp 1.0 for Tivoli 4.1  Pristine Manager Gateway, Version 4.3.1  Tivoli Management Framework SSLA Version 1.3  Resource Manager Gateway, Version 4.3.1  Distribution Status Console, Version 4.3.1  Software Distribution Gateway, Version 4.3.1  Patches:  Tivoli Framework Patch 4.3.1-LCF-0002 (build 09/23)  Scalable Collection Service, Version 4.3.1, Fix Pack 4.3.1-TIV-CLL-FP0001 (U829862 - 2009/10)  Inventory Gateway, Version 4.3.1, Fix Pack 4.3.1-TIV-INVGW-FP0001 (U829862 - 2009/10)  Software Distribution Gateway, Version 4.3.1, Fix Pack 4.3.1-TIV-SWDGW-FP0001 (U829862 - 2009/10)  Tivoli Framework Patch 4.3.1-TMF-0005 (build 11/23)  Tivoli Management Framework SSLA Version 1.3  Scalable Collection Service, Version 4.3.1  The comprehensive product and patch configuration files tmr.cfg and gw.cfg are located in /usr/local/Tivoli/etc/inst. These config files define the full product and patch configuration for TMR server or gateway managed node. The scripts reload-tmr and reload-gw (located under /usr/local/Tivoli/etc/script /) will run through the full configuration files by default, re-installing all products and patches, with no other options.  #Tivoli Management Framework Upgrade to Version 4.3.1 (build 11/19)  **GW Patches** (cm431-gw-upgd.cfg)  #Tivoli Management Framework Upgrade to Version 4.3.1 (build 11/19)  The comprehensive product and patch configuration files tmr.cfg and gw.cfg are located in /usr/local/Tivoli/etc/inst. These config files define the full product and patch configuration for TMR server or gateway managed node. This includes all products and patches from USDA Forest Service CM423 TMR Install as well as those introduced by this upgrade. The scripts reload-tmr and reload-gw (located under /usr/local/Tivoli/etc/script/) will run through the full configuration files by default with no other options. | | | |
| Prerequisites | | | |
| **Files**  usdafs\_cm431\_1of3.tar (1.6 GB)  usdafs\_cm431\_2of3.tar (1.6 GB)  usdafs\_cm431\_3of3.tar (1.5 GB)  **Environment**  TMR server running AIX v5.3 operating system  TMR server full system backup performed  GW managed node servers running AIX v5, RHEL v5 orSLES10 operating system  Endpoint clients running Windows, AIX v5, RHEL v5 or SLES10 operating systems  **Requirements**  root administrative privileges on all servers  Oracle 10g database client installed on TMR server  TMR server minimum of 5GB disk space available under filesystem /usr/local/Tivoli  TMR server minimum of 15GB disk space available under filesystem /usr/local/Tivoli/src  TMR server minimum of 10GB disk space available under filesystem /usr/local/Tivoli/depot | | | |

SECTION I: Manual Install Instructions  
Command prompt for root is designated by the pound symbol (#).

1. Log on to TMR server
   1. Start terminal session on Tivoli Management Region (TMR) server as root user. Confirm root user logged into TMR server.

**# hostname**

**# whoami**

1. Create directories
   1. Create following directories with at least the minimum of free disk space available. It is recommended, but not required to create filesystems for the directories.

Directory Min Size Recommended Size

/usr/local/Tivoli 5GB 5GB

/usr/local/Tivoli/src 15GB 20GB

/usr/local/Tivoli/depot 10GB 20GB

**Create directories on TMR server.**

1. Transfer files
   1. Copy usdafs\_cm431\_1of3.tar, usdafs\_cm431\_2of3.tar and usdafs\_cm431\_3of3.tar and checksum files to /usr/local/Tivoli/depot. The directory /usr/local/Tivoli/depot or any sufficiently large directory may be used to temporarily store the tar files. After installation the tar files should be removed.

**Copy install files usdafs\_cm431\_1of3.tar, usdafs\_cm431\_2of3.tar and usdafs\_cm431\_3of3.tar from DVD or network to TMR server /usr/local/Tivoli/depot.**

1. Extract files

Verify tar file and extract contents then extract src. The checksum of the tar files are included. The checksum of the tar file must be identical to the contents of the .cksum file. If not the file may be corrupt.

* 1. Change directory to /usr/local/Tivoli/.

**# cd /usr/local/Tivoli**

* 1. Run command to untar files under directory /usr/local/Tivoli/. If disk space is scarce, each tar file may be untarred manually, after which the tarfile may be deleted.

**# tar xvpf ./depot/usdafs\_cm431\_1of3.tar**

**# tar xvpf ./depot/usdafs\_cm431\_2of3.tar**

**# tar xvpf ./depot/usdafs\_cm431\_3of3.tar**

* 1. While still in directory /usr/local/Tivoli/. Untar etc directory.

**# tar xvpf ./src/431etc.tar**

* 1. Run script crtivdir to create additional directories and set permissions. This script was tested for AIX v5 servers only.

**# /usr/local/Tivoli/src/crtivdir**

The script will create directories that don't exist and set permissions as per following chart:

Directory Mode

/usr/local/Tivoli/ 2775

/usr/local/Tivoli/src/ 2775

/usr/local/Tivoli/depot/ 2777

/usr/local/Tivoli/etc/ 2775

/usr/local/Tivoli/var/ 2777

/usr/local/Tivoli/tmp 2777

/usr/local/Tivoli/log/ 2777

/usr/local/Tivoli/archive/ 2777

/usr/local/Tivoli/log/cm431 2777

/usr/local/Tivoli/db\_bkup 2775

/usr/local/Tivoli/backup 2777

/usr/local/Tivoli/etc/cfg 2775

1. Environment file

The following environmental variables used in this procedure. These variables are set in the file .tivenv and are included for expediency.

ult=/usr/local/Tivoli

ultv=/usr/local/Tivoli/var

ults=/usr/local/Tivoli/src

ulte=/usr/local/Tivoli/etc

ultes=/usr/local/Tivoli/etc/script

ultec=/usr/local/Tivoli/etc/cfg

lcfdir=/usr/local/Tivoli/lcf/dat/1

logdir=/usr/local/Tivoli/var/log

* 1. Source custom environmental variable file. Verify that variable $ult is set to /usr/local/Tivoli. Verify the command returns /usr/local/Tivoli.

**# . /usr/local/Tivoli/etc/cfg/.tivenv**

**# echo $ult**

1. Configuration files

This build process uses a set of region-specific configuration files used during and after the installation. A description of pertinent configuration files is included. These files may be prepared well in advance.

It may be helpful to put together a list of managed nodes to be created in entire region. Then decide on a gateway label for node.

Preparation of some configuration files requires managed node or gateway labels (mdist.cfg and epmgr.cfg). The managed nodes and gateways haven't been created yet, so these configuration file will have to be checked against actual values before implementation.

Hub TMR: with no managed nodes or endpoints, only struct-subs.cfg needs to be prepared.

* 1. Policy Region and Profile Manager Configuration
     1. Configuration file defines custom policy regions and profile managers for endpoint subscriptions. Change to directory /usr/local/Tivoli/etc/cfg/struct.

**# cd $ultec/struct**

View file /usr/local/Tivoli/etc/cfg/struct/struct-subs.cfg-template for example of syntax. Each line of struct-subs.cfg should have a pair defining the policy region/profile manager to be created with parent policy region/profile manager. The defined policy regions and profile managers will be created in a later step (using make-reg). The PRs and PMs will be created in the order they are defined.

Example struct-subs.cfg

#Child Parent

r2.subs.pr r2.pr

r2.subs.RMRS.pr r2.subs.pr

r2.subs.RMRS.svr-aix-all.pm r2.subs.RMRS.pr

r2.subs.RMRS.HQ.svr-aix.dpm r2.subs.RMRS.svr-aix-all.pm

r2.subs.RMRS.RapidCityFSL.svr-aix.dpm r2.subs.RMRS.svr-aix-all.pm

r2.subs.ArapRoosNF.pr r2.subs.pr

r2.subs.ArapRoosNF.svr-aix-all.pm r2.subs.ArapRoosNF.pr

r2.subs.ArapRoosNF.SO.svr-aix.dpm r2.subs.ArapRoosNF.svr-aix-all.pm

r2.subs.ArapRoosNF.BoulderRD.svr-aix.dpm r2.subs.ArapRoosNF.svr-aix-all.pm

r2.subs.ArapRoosNF.dsk-wnt-all.pm r2.subs.ArapRoosNF.pr

r2.subs.ArapRoosNF.SO.dsk-wnt.dpm r2.subs.ArapRoosNF.dsk-wnt-all.pm

r2.subs.ArapRoosNF.BoulderRD.dsk-wnt.dpm r2.subs.ArapRoosNF.dsk-wnt-all.pm

The naming convention consists of a the following components:

Policy regions: <region>.subs.<forest>.pr

Profile managers: <region>.subs.<forest>.<interp>.pm

Dataless profile managers: <region>.subs.<forest>.<station>.<interp>.dpm

Where <region> is same region label the TMR will be created with, <forest> is Forest name, <station> is a sub-division of Forest, and <interp> is the interpreter type.

* + 1. Create file struct-subs.cfg defining custom policy regions and profile managers. The region and profile manager names should align with the actual TMR name as output by Framework wtmrname command.

**Create and populate file struct-subs.cfg**

* 1. Endpoint Policy Configuration

If no action is taken, the included endpoint validation policy scripts will be used. If custom scripts are preferred, they may be put the policy\_ep directory in place of the default scripts.

* + 1. Change to directory /usr/local/Tivoli/etc/cfg/policy\_ep.

**# cd $ultec/policy\_ep**

* + 1. The following endpoint validation policy files are included. Customize or replace files as needed. Endpoint policy will be set at a later step using script set-policy\_ep.

after\_install\_policy

allow\_install\_policy

login\_policy

select\_gateway\_policy

* 1. Endpoint Policy after\_install Configuration

Define the dataless profile managers to which endpoints are subscribed upon initial login, based on network address.

* + 1. Change to directory /usr/local/Tivoli/etc/cfg.

**# cd $ultec**

* + 1. Create file after\_install.cfg (used by after\_install\_policy endpoint policy). See after\_install.cfg-template for an example.

Example after\_install.cfg

#Subnet Interp ProfileManager #Comment

9.99.15.\* w32-ix86 <region>.subs.RO1.dsk-wnt.dpm #comments

9.99.15.\* linux-ix86 <region>.subs.RO1.svr-linux.dpm

9.99.15.\* aix4-r1 <region>.subs.RO1.svr-aix.dpm

9.99.199.\* w32-ix86 <region>.subs.RO2.dsk-wnt.dpm #comments

9.99.199.\* linux-ix86 <region>.subs.RO2.svr-linux.dpm

9.99.199.\* aix4-r1 <region>.subs.RO2.svr-aix.dpm

\*.\*.\*.\* w32-ix86 <region>.subs.RO.dsk-wnt.dpm #tmr catch-all

\*.\*.\*.\* linux-ix86 <region>.subs.RO.svr-linux.dpm #tmr catch-all

\*.\*.\*.\* aix4-r1 <region>.subs.RO.svr-aix.dpm #tmr catch-all

Each line consists of a the following components:

<subnet> <interp> <profile\_manager> #comment

Where <subnet> may consist of wild-card elements (\*), which will be matched from right to left.

**Create and populate file after\_install.cfg**

* 1. Endpoint Policy allow\_install and select\_gateway\_policy Configuration

Define gateways to be assigned to an endpoint based on network address.

* + 1. Change to directory /usr/local/Tivoli/etc/cfg.

**# cd $ultec**

Create epmgr.cfg file with default gateways defined for a given subnet. View file epmgr.cfg-template for example of syntax. Endpoints originating from a specified subnet will be logged in to the corresponding gateway.

In following example, substitute actual gateway label for <gwX\_label> and TMR gateway label for <tmr-gw> (as returned by the wgateway command). This file must be checked for correspondence with actual gateway labels after gateway creation.

Example epmgr.cfg

#Subnet Gateway Enable(y/n)

99.1.1.\* <gw1\_label> y

99.1.\*.\* <gw2\_label> y

\*.\*.\*.\* <tmr-gw> y

The last line of epmgr.cfg allows the TMR server gateway to be defined as default gateway if no other is defined.

**Create and populate file epmgr.cfg**

* 1. Gateway Repeater Configuration

Define repeater parameters for each managed node.

* + 1. Change to directory /usr/local/Tivoli/etc/cfg.

**# cd $ultec**

* + 1. Create mdist.cfg file with repeater configuration (set via wrpt and wmdist) on one line for each managed node, comma-separated. View the file mdist.cfg-template for example of syntax.

The <GW\_dispatcher> and <client\_range> values are not assigned at this stage. The managed node labels can be defined on each line, but dispatcher values should be left unfilled, since these can only be known after managed nodes and gateways are created.  
In following example, substitute actual gateway label for <TMR\_managed\_node\_label> and <GW\_managed\_node\_label>. The <client\_range> field defines the repeater clients and may be specified as a single number, a colon separated-list (1:5:34:101) or a continuous range (5-25).  
Example mdist.cfg

# MN, OID, wan, range, mem\_max, disk\_max, disk\_hiwat, disk\_time, disk\_dir, net\_load, max\_conn, stat\_intv, rpt\_dir, permanent\_storage, max\_sessions\_high, max\_sessions\_medium, max\_sessions\_low, disk\_max, mem\_max, send\_timeout, execute\_timeout, notify\_interval, conn\_retry\_interval, retry\_ep\_cutoff, net\_load, packet\_size, target\_netload, debug\_level

<TMR\_managed\_node\_label>, 1, wd-, <client\_range>, 128000, 4800000, 4800000, 1, /usr/local/Tivoli/depot, 5000, 100, 180, /usr/local/Tivoli, TRUE, 5, 5, 5, 4800, 128, 300, 600, 10, 900, 7200, 5000, 16, 0, 3

<GW\_managed\_node\_label>, <GW\_dispatcher>, ,<client\_range> , 128000, 4800000, 4800000, 1, /usr/local/Tivoli/depot, 5000, 100, 180, /usr/local/Tivoli, TRUE, 5, 5, 5, 4800, 128, 300, 600, 10, 900, 7200, 5000, 16, 0, 3

<GW\_managed\_node\_label>, <GW\_dispatcher>, ,<client\_range> , 128000, 4800000, 4800000, 1, /usr/local/Tivoli/depot, 5000, 100, 180, /usr/local/Tivoli, TRUE, 5, 5, 5, 4800, 128, 300, 600, 10, 900, 7200, 5000, 16, 0, 3

The labels, dispatchers, and range should be unique on each line, but the other example values may be used as base values.

**Create and populate file mdist.cfg**

1. Create TMR server

Run Tivoli provided pre-installation script then create TMR server. Install products and patches, then configure. From TMR server, initiate SSH connection to TMR server, specifying TMR fully qualified domain name (FQDN) as target. This step stores the TMR public key in the SSH known\_hosts file, and allows endpoint installation to proceed in script insttiv. Substitute actual TMR server FQDN for <TMR FQDN>.

**# ssh root@<TMR FQDN>**

* 1. Change directory to /usr/local/Tivoli/etc/cfg/inst. Prepare TMR configuration files for a fresh TMR build.

**# cd $ultec/inst**

**# cp tmr-fresh.cfg tmr.cfg**

* 1. Create base Framework v4.1.1 TMR server, install products/patches, create gateway and create endpoint. This operation runs the Tivoli WPREINST.SH and wserver scripts. This process supports AIX TMRs only.

Product and patch installation in this operation use custom configuration files and the Tivoli winstall and wpatch commands.

Change to log file directory /usr/local/Tivoli/var/log/cm431/. Supply actual TMR server root password for <password>. If the -p option and argument are not supplied, the password will be requestd as user input. Substitute desired TMR region label for <region\_label>. Example region labels are r1, r2, mci and abq. The TMR region name will be set to '<region\_label>.pr' which is the value returned by Framework command wtmrname

**# cd $logdir/cm431**

**# $ultes/insttiv -p <password> -r <region\_label> 2>&1 | tee insttiv.log**

* 1. Set environment. Add the following three lines at end of the root user's .profile if not already present.

**if [ -f /etc/Tivoli/setup\_env.sh ]; then**

**. /etc/Tivoli/setup\_env.sh**

**fi**

* 1. Log out root user and then log in again to source the environment file.

**# exit**

* 1. Once logged in again as root user, verify Tivoli framework environment. Print the defined value of the environmental variable $DBDIR.

**# echo $DBDIR**

The previous command should have returned the path to the Tivoli object database directory. If nothing was returned, check that the Tivoli environment file /etc/Tivoli/setup\_env.sh exists and confirm the lines added previously to the .profile.

* 1. Verify Tivoli root administrator roles. The Tivoli root administrator should have all global roles.

**# wgetadmin | grep ^roles**

Example of Tivoli administrator with all global roles.

Global TMR Authorization Roles

roles: global super, senior, admin, user, backup, restore, install\_client, install\_product, policy, Query\_view, Query\_execute, Query\_edit, RIM\_view, RIM\_update, HTTP\_Control, Dist\_control, Inventory\_view, Inventory\_scan, Inventory\_edit, Inventory\_end\_user, APM\_Manage, APM\_View, APM\_Edit, APM\_Admin, CCM\_View, CCM\_Manage, CCM\_Edit, CCM\_Admin, WebUI\_Admin, Query\_directory\_view, Query\_directory\_execute, Query\_directory\_edit, Pristine\_Read, Pristine\_Write, Pristine\_Execute

* 1. Change to directory /usr/local/Tivoli/var/log/cm431. Run script to create TMR gateway, policy regions and profile managers and set ownership and permisions for files and directories.

**# cd $logdir/cm431**

**# $ultes/config-tmr 2>&1 | tee config-tmr.log**

1. Create Gateway managed nodes

Hub TMR: skip this step.

Create Tivoli managed node and install endpoint gateway, products, patches and endpoint to create Tivoli Gateway server. Supported platforms include AIX v5, RHEL v5 and SLES10.

As root user on TMR server, confirm current directory is /usr/local/Tivoli/var/log/cm431, the common log repository for this process.

**REPEAT THIS STEP FOR ALL GATEWAY SERVERS**

* 1. Create Tivoli managed node and endpoint gateway on target server.

Install products and patches on managed node. This operation uses custom configuration files and the Tivoli wclient, winstall and wpatch commands. Substitute the actual managed node label for <managed\_node> (this is the <fqdn> value used in the managed node creation). Managed nodes can be listed by running command 'wlookup -aLr ManagedNode'.

Run script to install managed node, endpoint gateway, products and patches. Where <fqdn> is fully qualified domain name of target, and <gateway\_label> is label of gateway object to be created. The gateway label should end with -gw. The managed node will be created using <fqdn> as label.

IMPORTANT: The -j option uses SSH protocol to connect to target. Prior to installation, an SSH connection must be initiated from TMR server to target host, specified by FQDN. This stores the target host's public key in the ~/.ssh/known\_hosts file. If host is not in known\_hosts file, the operation will time out with a host authentication error.

**# $ultes/inst-mn -j -p <password> <fqdn> <gateway\_label> 2>&1 | tee inst-mn.<fqdn>**

Review log file for errors. Retry managed nodes that experience failures.

* 1. Create endpoint

Create endpoint on Gateway server using SSH. A SSH connection must be initiated to the target host prior to installing the endpoint, as was performed in prior gateway managed node install step. Where <password> is root password and <fqdn> will be set as the endpoint label.

**# $ultes/inst-ep -j -p <password> <fqdn> 2>&1 | tee inst-ep.<fqdn>**

Managed node created with endpoint gateway, products/patches and endpoint installed. Proceed to next Gateway installation.

1. Inventory database schema  
   Install Tivoli inventory database schema on Oracle database.
   1. Prepare database admin and schema scripts.

The database administrator should be provided with these scripts beforehand in order to customized the datafiles specifications. Original database scripts are located at /usr/local/Tivoli/src/cm431install/SQL.

Oracle DBA must edit admin scripts to specify location, name and size of datafiles (.dbf) and extent values for each tablespace definition. If multiple databases are running on an Oracle RAC the fully qualified file names must be uniqure. If multiple tablespaces specify the same fully qualified file, corruption will ensue.

The following default USDA Forest Service TMR database admin and schema scripts are included in the directory /usr/local/Tivoli/etc/script/sql/SQL/admin.

fs\_inv\_ora\_admin.sql

fs\_ccm\_ora\_admin.sql

fs\_mdist\_ora\_admin.sql

fs\_plans\_ora\_admin.sql

fs\_pristine\_ora\_admin.sql

* + 1. Change ownership and permissions to oracle.dba on sql directory. Create log directory for sql script results.

**# chown -R oracle.dba $ultes/sql**

**# mkdir -p -m775 $logdir/cm431/sqllog**

* + 1. Change to user oracle.

**# su - oracle**

* + 1. Source custom environment file .tivenv.

**$ . /usr/local/Tivoli/etc/cfg/.tivenv**

* 1. Run SQL ADMIN scripts
     1. Change to database script logging directory /usr/local/Tivoli/var/log/cm431/sqllog.

**$ cd $logdir/cm431/sqllog**

The following admin SQL scripts will be executed. Scripts are located at /usr/local/Tivoli/etc/script/sql/SQL/admin/. Note all admin script will be run "as sysdba".

Script User Password

fs\_inv\_ora\_admin.sql sys usfsora

fs\_ccm\_ora\_admin.sql sys usfsora

fs\_mdist\_ora\_admin.sql sys usfsora

fs\_plans\_ora\_admin.sql sys usfsora

fs\_pristine\_ora\_admin.sql sys usfsora

* + 1. While still in SQL log directory /usr/local/Tivoli/var/log/cm431/sqllog/, run CM431 inventory admin scripts as Oracle user sys and password <sys\_password>. Substitute the actual database net service name for <db\_name>.

**$ $ultes/sql/tiv\_schema\_setup -s1 -p <sys\_password> <db\_name> 2>&1 | tee -a admin.log**

Review log files for errors. Expect to see a set of errors that are non-catastrophic, for example, drop actions just prior to create actions.

* 1. Run SQL SCHEMA scripts

The following schema SQL scripts will be executed. Scripted are located at /usr/local/Tivoli/etc/script/sql/SQL/schema/.

Script User Password

fs\_inv\_ora\_schema.sql invtiv tivoli

fs\_ccm\_ora\_schema.sql tivoli tivoli

fs\_mdist\_ora\_schema.sql mdstatus mdstatus

fs\_plans\_ora\_schema.sql planner planner

fs\_pristine\_ora\_schema.sql pristine pristine

* + 1. While still in SQL log directory /usr/local/Tivoli/var/log/cm431/sqllog/, run CM431 inventory schema scripts. Substitute the actual database net service name for <db\_name>.

**$ $ultes/sql/tiv\_schema\_setup -s2 <db\_name> 2>&1 | tee -a schema.log**

Review log files for errors. Expect to see a set of errors that are non-catastrophic. For example, drop actions just prior to create actions.

* 1. Run SQL SCHEMA FIXPACK scripts

The following schema fixpack SQL scripts will be executed. Scripted are located at /usr/local/Tivoli/etc/script/sql/RDBMS/.

Script User Password

fs\_inv\_ora\_schema\_431\_FP01.sql invtiv tivoli

* + 1. While still in SQL log directory /usr/local/Tivoli/var/log/cm431/sqllog, run CM431 inventory schema fix pack scripts. Substitute the actual net service database name for <db\_name>.

**$ $ultes/sql/tiv\_schema\_setup -s3 <db\_name> 2>&1 | tee -a fixpack.log**

Review log files for errors. Expect to see a set of errors that are non-catastrophic. For example, drop actions just prior to create actions.

* 1. Exit user Oracle and return to root session.

**$ exit**

1. TMR Configuration

Configure Tivoli Management Region components. Create database interfaces, run scripts to configure Inventory, Software Distribution, Change Configuration Manager, Activity Planner and Resource Manager.

On TMR server as user root. The environment file /usr/local/Tivoli/etc/cfg/.tivenv must be sourced.

* 1. Create RIM objects

The following RIM objects will be created:

RIM User Password

inv\_query invtiv tivoli

invdh\_1 invtiv tivoli

mdist2 mdstatus mdstatus

planner planner planner

ccm tivoli tivoli

trm tivoli tivoli

* + 1. Change directory to /usr/local/Tivoli/var/log/cm431.

**# cd $logdir/cm431**

* + 1. As root user, run script to create Tivoli RIM objects on TMR server for communication with Oracle database. The location of $ORACLE\_HOME is expected to be /var/lpp/oracle. Where <net\_service\_name> is database net service name, and <instance\_name> is database instance name. The variables net\_service\_name and instance\_name are defined in /var/lpp/oracle/network/admin/tnsnames.ora.

**# $ultes/create-rim -n <net\_service\_name> -i <instance\_name> 2>&1 | tee create-rim.log**

* + 1. Run script to verify RIM objects. Upon successful RIM connection, the session is released with an x to exit, and the next RIM object is tested.

**# $ultes/tool/rim\_test.sh**

If command returns a connection error, then run wgetrim for RIM object to get its attributes and check the net service name and instance name.

* 1. Configure Inventory
     1. While still in directory /usr/local/Tivoli/var/log/cm431, run script to configure Tivoli Inventory v4.3.1 and import software signatures.

**# $ultes/inv/config-inv 2>&1 | tee config-inv.log**

Latest software signature file, located at $ultec/inv/ITLCM23\_20100131/itlcm22-allProducts-fullSwCat-20100131.xml, is imported. The command winvsig is deprecated. The command winvmigrate should be used to import signatures from newer xml software catalogs. The latest software signature file (IBM Software Catalog for ITLCM 2.2/2.3 - All Software) may be downloaded from the URL below.

<http://www-306.ibm.com/software/sysmgmt/products/support/IBMTivoliLicenseManager.html>

Note: After successfully processing software signatures, error message "INVCO0294I No valid package was found." will be displayed, because no package was migrated. This error can be disregarded.

* 1. Configure Activity Planner
     1. While still in directory /usr/local/Tivoli/var/log/cm431, run script to configure Tivoli Activity Planner v4.3.1. Confirm that tivapm user account is accessible by logging onto system as tivapm. Any login restrictions (such as expired password, failed logins, etc.) must be removed before moving on. The Activity Planner cannot start until tivapm is available.

**# $ultes/apm/config-apm 2>&1 | tee config-apm.log**

* + 1. Set Tivoli activity planner password. Use tivapm for the old and new password.

Old password: tivapm

New password: tivapm

**# wsetapmpw**

* 1. Migrate TMR endpoint

Hub TMR: Delete endpoint instead.

* + 1. Migrate TMR endpoint from TMR gateway to some other gateway. The choice of gateway is arbitrary, so long as it is not the TMR gateway. Where <TMR\_endpoint> is TMR endpoint label and <gateway\_label> is gateway label.

**# wep <TMR\_endpoint> migrate <gateway\_label>**

* 1. Configure endpoint validation policy

Hub TMR: Skip this step.

* + 1. Update endpoint validation policy configuration file to reflect actual gateway labels and subnets.

Confirm values in file /usr/local/Tivoli/etc/cfg/epmgr.cfg are correct. This file was manually populated in an earlier step. Verify gateways specified in config file correspond to actual gateway labels. Ensure third field (column) is set to 'y' to enable subnet.

**Verify file /usr/local/Tivoli/etc/cfg/epmgr.cfg.**

* + 1. Run script to configure TMR endpoint validation policy.

**# $ultes/set-policy\_ep**

* 1. Configure gateway repeaters

Hub TMR: Skip this step.

* + 1. Update repeater configuration file to reflect actual manged node labels and dispatcher values. Update /usr/local/Tivoli/etc/cfg/mdist.cfg config file with actual <TMR\_dispatcher>, <GW\_dispatcher> and <client\_range> values. The <client\_range> field defines the repeater clients and may be specified as a single number, a colon separated-list (1:5:34:101) or a continuous range (5-25). Also verify all managed node labels are correct.

**Update the /usr/local/Tivoli/etc/cfg/mdist.cfg config file with actual <TMR\_dispatcher>, <GW\_dispatcher> and <client\_range> values.**

* + 1. While still in directory /usr/local/Tivoli/var/log/cm431, run script to configure the gateway repeater wmdist and wrpt parameters. Both scripts use configuration file /usr/local/Tivoli/etc/cfg/mdist.cfg to define repeater parameter values.

**# $ultes/tune-mdist 2>&1 | tee tune-mdist.log**

**# $ultes/tune-rpt 2>&1 | tee tune-rpt.log**

1. Restart object dispatchers

Restart all nodes to enable full functionality.

* 1. Shutdown all client managed node object dispatchers. Restart TMR server object dispatcher. Start all client managed node object dispatchers.

**# odadmin shutdown clients**

**# odadmin reexec**

**# odadmin start clients**

1. Inter-region connection

Connect TMR server to TEC server for exchange of event data. A two-way connection will be initiated.

* 1. While still logged into TMR server as root, connect TMR server to TEC server.

**# connect-tec TEC\_host 2>&1 | tee connect-tec.log**

Where TEC\_host is TEC server hostname.

* 1. Confirm TMR-TEC connection. Run wlsconn to get a regional connection listing.

**# wlsconn**

Hub TMR: Connect HUB TMR to spoke TMRs with wconnect.

1. Install Tivoli Desktop for Windows

The Tivoli Desktop for Windows client provides a GUI interface to the TMR server. To install the Tivoli Desktop for Windows GUI copy Config\_Mgr\_V431\_desktop.zip and cm431fp01\_desktop.zip from TMR server to Windows workstation.

* 1. Copy files from TMR server to Windows workstation. The two files combined need 1GB disk space.

**Copy /usr/local/Tivoli/src/Config\_Mgr\_V431\_desktop.zip to Windows client.**

**Copy /usr/local/Tivoli/src/cm431fp01\_desktop.zip to Windows client.**

* 1. Extract both zip files on Windows client under any directory, for example c:\tmp.

**Extract files Config\_Mgr\_V431\_desktop.zip and cm431fp01\_desktop.zip**

* 1. Change directory to where zip files were extraced in previous step, for example c:\tmp. From there change directory again to Config\_Mgr\_V431\_desktop. Run setup.exe to install Tivoli Desktop for Windows v4.3.1.

**Run setup.exe**

* 1. Upgrade Desktop

Change directory to where zip files were extracted in previous step, for example c:\tmp. From there change directory again to cd2\_cm431fp01\cd2\spb\_installer. Run Setup.exe to upgrade to Tivoli Desktop for Windows v4.3.1 Fix Pack 1.

When prompted, specify cd2\_cm431fp01\cd2\package\CM431\_SPB\_FP01.xml as XML configuration file.

**Run Setup.exe**

1. Check and backup object database

All TMR installation and configuration has been completed. Check and backup Tivoli object database.

* 1. Check Tivoli object database and fix errors. Run the wchkdb command two times as indicated. The second run should have a clean result.

**# wchkdb -u**

**# wchkdb -u**

* 1. Perform Tivoli object database backup. Substitute actual region label for <region>.

**# wbkupdb -sd $ultv/db\_bkup/<region>-db\_bkup-post\_cm431**

SECTION II: Tivoli Install Instructions

N/A

SECTION III: Validation Steps

1. Validate TMR Framework level is 4.3.1.

**# wlsinst –a |grep Framework**

1. Validate TMR Configuration Manager level is 4.3.1.

**# wlsinst –a | grep Inventory**

1. Validate TMR endpoint level is 43102.

**# wep <TMR\_endpoint\_label> get version**

SECTION IV: Backout Instructions

1. Restore TMR server from backup image.

Δ END OF Δ