Dynamic Provisioning of SAP Environments using IBM Dynamic Infrastructure for mySAP and IBM Tivoli Provisioning Manager

Introduction

This paper provides details about the dynamic provision of SAP application servers using IBM® Dynamic Infrastructure for mySAP.

The IBM Dynamic Infrastructure for mySAP solution relies on IBM Tivoli® Provisioning Manager V2.1 for the provisioning and deprovisioning operations.

This paper focuses on the tasks required for planning, implementing, and configuring all of the components that make up IBM Dynamic Infrastructure for mySAP. The following topics will be discussed:

- “IBM Dynamic Infrastructure for mySAP overview” on page 3
- “Integration scenario overview” on page 4
- “Configuring LDAP security” on page 14
► “IBM Dynamic Infrastructure for mySAP installation on the Tivoli Provisioning Manager server” on page 16
► “Verify installation of IBM Dynamic Infrastructure for mySAP components” on page 17
► “Tivoli Provisioning Manager Data Center modeling” on page 19
► “IBM Dynamic Infrastructure for mySAP configuration” on page 27
► “z/VM preparation and configuration” on page 33
► “SAP master systems setup” on page 43
► “Scenario execution” on page 52
IBM Dynamic Infrastructure for mySAP overview

IBM Dynamic Infrastructure for mySAP (IDI) is an IBM on demand solution for a heterogeneous environment that can enable clients to run SAP environments more efficiently by dynamic provisioning of SAP systems. It provides automated, policy-based, end-to-end management of resources, availability, accounting, and metering information about resource consumption across heterogeneous systems, driven by service level agreements. IBM Dynamic Infrastructure can:

- Reduce the high management effort for growing SAP landscapes
- Speed up the deployment of new SAP systems
- Improve the utilization of systems

One of the main tasks of IBM Dynamic Infrastructure for mySAP is to provision additional SAP dialog instance servers. It does this by taking servers from a pool of free servers and installing an SAP dialog instance server on it. At logon, new users are directed to that server, avoiding bottlenecks on the existing ones. The provisioning is done dynamically based on the response time defined in service level agreements. In detail, for the intent of this paper, the IBM Dynamic Infrastructure for mySAP solution provides the following functionality:

- **Dynamic provisioning**
  By allocating and de-allocating application server resources based on a predefined policy (for example, 80% of dialog steps should take less than one second), the system infrastructure shrinks and grows on demand. The utilization is increased, the deployment time is reduced, and at the same time the systems management is simplified.

- **Server consolidation**
  Sizing of the infrastructure is no longer oriented on peak workload. The usage of virtual servers (shared pools instead of dedicated resources) significantly reduces the investments in hardware and systems management.

- **Data security**
  De-provisioned servers are immediately scrubbed, thus ensuring that no customer data is transferred between environments. This is a requirement for hosting environments, and also has growing importance for internal service providers.

- **Increased average server utilization**
  IBM Dynamic Infrastructure for mySAP performs automatic resource requirements prediction based on predefined service level objectives and agreements, and automates infrastructure deployment. This ensures that each application has the resource it needs, when it needs it, without static overprovisioning.
Integration scenario overview

IBM Dynamic Infrastructure for mySAP relies on several components that must be installed and configured before any dynamic provisioning of application servers takes place.

The IBM Dynamic Infrastructure for mySAP solution has the following main components:

- An SAP base system consisting of:
  - An SAP Central Instance server
    This scenario uses an SAP Central Instance server running on SUSE SLES8 Linux® running on zSeries®.
  - A database server
    This scenario uses a DB2® database server running on z/OS®.
  - Various static application servers. The number of application servers depends on workload and agreed service level objectives.

- Infrastructure servers
  - SAP Dialog Instance Master server
    A server on which a reference installation for all Dialog Instances is performed and tested. This scenario uses an SAP Dialog Instance Master server running on SUSE SLES8 Linux running on zSeries.
  - SAP Configuration Master server
    A server on which the cloning configuration is maintained. This configuration is used during dynamic provision operations. This scenario uses an SAP Configuration Master server running on SUSE SLES8 Linux running on zSeries.
  - Managed-through server
    This server receives and executes tasks that assist in the provisioning operations. The managed-through server will be configured as a boot server in Tivoli Provisioning Manager. This scenario uses z/VM® to support the provisioning of Linux for zSeries servers.

- Management environment

  The management environment represents the environment that will manage the dynamic provisioning of SAP application servers. It consists of Tivoli Provisioning Manager and IBM Dynamic Infrastructure for mySAP software.

  The recommended topology used for the management environment is a two-server architecture as follows:
  - Server 1 running IBM DB2 Server and Tivoli Directory Server
– Server 2 running IBM WebSphere®, IBM DB2 Client, IBM Tivoli Provisioning Manager and prerequisite software, and IBM Dynamic Infrastructure for mySAP.

IBM Dynamic Infrastructure for mySAP running on the management server has the following components:

– Utility Business Services
  Responsible for accounting, metering, rating, and subscription services
– Metrics service
  Collects performance and workload measurement data from SAP systems. Data is passed on to the optimizer for analysis.
– Optimizer
  This element interacts with Tivoli Provisioning Manager and dictates the allocation and deallocation of new application servers based on current workload of customer environments.

► A set of heterogeneous servers to be used by the provisioning operations to deploy additional application servers, depending on demand.

These servers will be shared between the different customers’ SAP systems. In Tivoli Provisioning Manager terms, these servers will make up the resource pool of servers for the provisioning operations. Decisions about when to provision or deprovision servers from the resource pool to a customer’s SAP environment will be made by IBM Dynamic Infrastructure for mySAP based on the service level objectives and agreements defined for the customer.

The provisioning process is first triggered when a customer subscribes to an offering. Offerings have to be pre-configured in IBM Dynamic Infrastructure for mySAP and will be discussed in “IBM Dynamic Infrastructure for mySAP configuration” on page 27.

At a high level, as shown in Figure 1 on page 6, the following operations are performed at the time a subscription is being executed:
1. The customer accesses the IBM Dynamic Infrastructure for mySAP user interface to subscribe to an offering. In doing this, the customer chooses an offering and sets several service level agreements to its subscription.

2. The Utility Business Service component of IBM Dynamic Infrastructure for mySAP gathers and stores all property data connected with this subscription and generates a request to Tivoli Provisioning Manager to start the operations that set up and configure all resources needed by this subscription.

3. Tivoli Provisioning Manager then executes workflows and scripts that are provided by the IBM Dynamic Infrastructure for mySAP automation package, as well resources from other automation packages, depending on the target platform the SAP systems run on. In our case, the workflows create one or more new Linux on zSeries servers on our z/VM system, and install and configure an SAP application server on it.

For the subscription operation, the workflows create the necessary Data Center Model (DCM) infrastructure required for the subscription. The DCM entries are: Customer, Application, and Cluster definitions. The workflows proceed by provisioning servers to the defined cluster according to the details of the subscription. Tivoli Provisioning Manager allocates a server from the
designated resource pool and proceeds with the provisioning of all required software packages. After a successful provisioning operation, Tivoli Provisioning Manager marks the new server resource as used and moves its DCM representation from the resource pool to the Cluster definition.

4. The newly provisioned SAP application server is placed in the SAP logon group of the utilized SAP system and can be used by all users of this system. Tivoli Provisioning Manager then signals the Utility Business Services component that all workflows were executed successfully and the SAP application server is ready for use.

5. Utility Business Services now informs the customer that the subscription request has been completed and all resources associated to the offering are ready to use.

6. The Metrics service starts metering of this resource for later billing purposes.

7. The Optimizer then interacts with Tivoli Provisioning Manager and dictates the allocation and deallocation of new application servers based on current workload.

Existing IT infrastructure

In order to demonstrate the provisioning operations performed by IBM Dynamic Infrastructure for mySAP in this scenario, the following assumptions are made:

1. A functional management environment with IBM Tivoli Provisioning Manager V2.1 up and running.

2. A functional SAP base system infrastructure running on zSeries.

Figure 2 on page 8 shows the IT infrastructure used in this scenario. This is the start point for us to create a working IBM Dynamic Infrastructure for mySAP environment.
Figure 2  Initial IT infrastructure

Later in this paper, we show all of the tasks and configuration details that are required in the environment shown in the figure above to have the IBM Dynamic Infrastructure for mySAP environment operational.

There are several supported topologies for running IBM Dynamic Infrastructure for mySAP. Refer to IBM Dynamic Infrastructure Enterprise Edition for mySAP Business Suite Planning, BOEE-EUIPL-00 for a list of supported topologies.

The scenario presented in this paper uses the following topology, which in IBM Dynamic Infrastructure for mySAP terminology is referred to as topology 3:

- Management systems running on xSeries® with Microsoft® Windows® 2003 Server
- SAP application servers (provisioned servers in Figure 3 on page 13), SAP Dialog Instance, SAP Central Instance, SAP Configuration Master systems running SUSE SLES 8 Linux for zSeries
- Database server running on zSeries.

The following sections provide details of each component of the existing infrastructure, as illustrated in Figure 2.
SAP base system overview

Each customer has an SAP base system that consists of the database and core SAP processes instances required to run the SAP system. Each base system may be planned and deployed differently. The provisioning and deprovisioning of the SAP base system is outside of the scope of IBM Dynamic Infrastructure for mySAP.

The SAP base system must be up and running before any automated provisioning operation can occur. However, there are special considerations for the SAP Central Instance, SAP Dialog Instance, and SAP Configuration Master systems when considering IBM Dynamic Infrastructure for mySAP. Refer to IBM Dynamic Infrastructure Enterprise Edition for mySAP Business Suite Planning, BOEE-EUIPL-00 for details.

For the purpose of this scenario, SAP System 4.7 Enterprise Kit with kernel version 6.20 patch level 1211 is used. Also the SAP support pack 47 must be installed as required by IBM Dynamic Infrastructure for mySAP. The level of maintenance for your system will vary depending on the version of SAP.

The manuals recommend that the SAP Configuration Master hosts the SAP runtime that will be accessed via NFS by the provisioned SAP application servers. The SAP runtime must be accessible at all times any provisioned SAP application server is running. In our scenario, the SAP Configuration Master contains the SAP runtime locally, and as all the provisioned SAP application servers will be created as clones of the SAP Configuration Master server, the SAP runtime will be available to the SAP application servers locally as well.

Note: If your environment is not set up similarly or has multiple sets of cloned disks you may find it necessary or convenient to mount the file system containing the SAP runtime on each dynamically provisioned clone from the SAP Configuration Master.

The IBM Dynamic Infrastructure for mySAP solution has requirements around how the VM minidisks are defined for the SAP Configuration Master server. If the workflow defaults are used, the z/VM minidisks that are to be cloned must have DASD device addresses in the form of 1CXX to be recognized and copied during the provisioning operation. All z/VM minidisk DASD device addresses of the form 0CXX are the basis when creating the SAP Configuration Master server.

When the SAP Configuration Master server setup is complete, the content of the 0CXX minidisks must be copied to the 1CXX minidisks for cloning. All other DASD device addresses from 2XXX through FXXX are unused by the default provisioning operation. For details about how the SAP Configuration Master has been defined in our scenario, refer to “SAP master systems setup” on page 43.
As for the database server, IBM DB2 Server V7 for z/OS is running in an LPAR with z/OS V1.6 in our zSeries z990. Also, maintenance RSU 0502 and SAP service marketplace OSS 81737 are installed.

Because the version of SAP used in this scenario is 4.7, a started task (STC) named ICLIM5D was configured to serve as the Integrated Call Level Interface (ICLI) for the transfer of SQL statements from the SAP application servers to the DB2 database server.

**Note:** The version of SAP used will determine whether an ICLI or Distributed Relational Database Architecture™ (DRDA®) connection via DB2 Connect™ is required.

The SAP Agent component needs DDIC user privileges. In order to run, the SAP Agent needs certain SAP function modules installed on the monitored SAP systems. These function modules are provided by several SAP base support packages and are installed as a patch. In our environment as the SAP support pack 47 is installed, the monitored SAP servers must have the SAPKB62047 package installed.

The SAP Administrator has to make sure that all system collectors are running. The SAPOSCOL (running on the SAP monitored systems), RFCOSCOL (running on the database server), and all other collectors must be started automatically during startup.

IBM Dynamic Infrastructure for mySAP assumes that SAP users log on to a predefined SAP logon group. Each application server that belongs to the set of dynamically provisioned SAP application servers must be a member of a unique SAP logon group.

**Management environment overview**

The recommended topology used for the management environment is a two-servers architecture as shown in Figure 2 on page 8. The following list represents the software and hardware configuration on each of the servers:

- **Server one**
  - Cygwin DLL 1.5.14-1
  - IBM DB2 UDB Workgroup Server Unlimited Edition V8 with fix pack 3 and APAR IY57410 for DB2
  - IBM Tivoli Directory Server V5.2
Server two

- Cygwin DLL 1.5.14-1
- IBM WebSphere Application Server V5.1 with cumulative fix 3, IC38409 and MQ CSD05 fixes
- IBM DB2 UDB Workgroup Client Unlimited Edition, V8 with fix pack 3
- IBM Tivoli Provisioning Manager V2.1 with fix pack 1.

Refer to *IBM Tivoli Provisioning Manager V2.1 Installation Guide, GC32-1614* for details about setting up IBM Tivoli Provisioning Manager V2.1. For Tivoli Provisioning Manager fix pack 1, refer to:


IBM Dynamic Infrastructure for mySAP requires the SAP Java™ Connector (JCo)—specifically, the Metrics and Optimizer service. The JCo is a collection of Java classes and system-dependent runtime libraries that provide RFC functionality for Java applications.

The JCo installation package is not part of the IBM Dynamic Infrastructure for mySAP installation media. JCo is available from the SAP Service Marketplace and can be obtained from http://service.sap.com/connectors. When authenticated with an SAP Service account, from the main menu, select Software download, select Search for all categories, enter JCo and select Search. Choose the appropriate code from the list that appears.

IBM Dynamic Infrastructure for mySAP requires additional software and fixes to be installed on the management environment. This is discussed in “IBM Dynamic Infrastructure for mySAP installation on the Tivoli Provisioning Manager server” on page 16.

Additional automation packages will be installed to Tivoli Provisioning Manager by the IBM Dynamic Infrastructure for mySAP installation process. These automation packages contain workflows, Java plug-ins, logical operations, and external scripts necessary to carry out the provisioning operations in an IBM Dynamic Infrastructure for mySAP environment. The installation verification of these automation packages is discussed in “Verify installation of IBM Dynamic Infrastructure for mySAP components” on page 17.

Additional elements must be defined in the Tivoli Provisioning Manager Data Center Model (DCM) for the SAP infrastructure utilized by IBM Dynamic Infrastructure for mySAP. These elements are presented in “Tivoli Provisioning Manager Data Center modeling” on page 19.
Network overview
As stated in IBM Dynamic Infrastructure Enterprise Edition for mySAP Business Suite Planning, BOEE-EUIPL, several LANs should be set up for optimum performance security and availability. The recommended LANs are ADMINLAN, CUSTLAN, and DBLAN. As Tivoli Provisioning Manager requires the use of VLANS, and to keep our scenario as simple as possible, we chose to combine all LANs described above into a single VLAN.

In a production environment it is strongly recommended that several LANs (or VLANs) should be set up to enhance overall network performance and security.

To keep the Linux on zSeries network setup and configuration simple and to ensure minimal system changes after the provisioning operations complete, we chose to make use of a vswitch in z/VM. This enabled definition of the SUSE SLES 8 Linux for zSeries servers with a virtual device address for the network that could be duplicated for each and every additional system we needed to provision, without having to perform additional customization work.

Integration implementation overview

The goal of this paper is to implement an environment similar to Figure 3 on page 13.
To implement the pictured scenario, the following tasks must be performed:

- Configure LDAP security.
- Install IBM Dynamic Infrastructure for mySAP on the Tivoli Provisioning Manager server.
- Configure the Tivoli Provisioning Manager Data Center Model.
- Configure IBM Dynamic Infrastructure for mySAP.
- Configure the z/VM environment.
- Set up the SAP master systems.

The above tasks will be presented in details in the following sections.
Configuring LDAP security

IBM Dynamic Infrastructure for mySAP relies on IBM WebSphere and Tivoli Directory Server for authentication and authorization services, and requires the definition of various groups and user IDs.

Users first provide credential information using a login panel. IBM WebSphere checks the user’s credentials against definitions in LDAP. The user’s rights will be based on the groups the user is assigned to.

Users and groups definitions for IBM Dynamic Infrastructure for mySAP can be stored in the same LDAP server used for Tivoli Provisioning Manager. This can be performed as follows:

- User ID definitions using the Tivoli Provisioning Manager graphical user interface
- Group definitions using a Lightweight Directory Interchange Format (LDIF) file

User ID definitions

The definitions of user IDs must be planned for prior to IBM Dynamic Infrastructure for mySAP installation. For our case study scenario, we defined the following user IDs: dexteradmin, dexterservicemanager, dexterodadmin, dexteremea, dexterap, and dexteramericas.

To define the user IDs in Tivoli Provisioning Manager, perform these steps:

1. On the Tivoli Provisioning Manager user interface, access the Systems configuration and workflows management tab and select Users.
2. On the right panel, select Edit → Add User. Fill out the required fields.

Group definitions

IBM Dynamic Infrastructure for mySAP uses the following four groups:

- OfferingProviderAdministrator
- OfferingServiceManager
- OfferingSubscriber
- ODSProviderAdministrator

For details about each group’s privileges, refer to IBM Dynamic Infrastructure Enterprise Edition for mySAP Business Suite Installation and Customization, BOEE-EUIN-00.

After the LDIF file is created, its definitions can be loaded into the LDAP database by either specifying the location of the file during the IBM Dynamic
Infrastructure for mySAP installation, or import the file using the `ldapmodify` command.

The following example shows the LDIF file used in this scenario. It also shows the user ID and group membership relationships used in this scenario.

**Example 1   LDAP group definitions for IBM Dynamic Infrastructure for mySAP**

```plaintext
dn: cn=OfferingProviderAdministrator,dc=ibm,dc=com
  cn: OfferingProviderAdministrator
  objectClass: groupOfUniqueNames
  description: Offering Provider Administrator
  uniquemember: cn=dexteradmin,dc=ibm,dc=com

dn: cn=OfferingServiceManager,dc=ibm,dc=com
  cn: OfferingServiceManager
  objectClass: groupOfUniqueNames
  description: Offering Service Manager
  uniquemember: cn=dexterservicemanager,dc=ibm,dc=com

dn: cn=ODSProviderAdministrator,dc=ibm,dc=com
  cn: ODSProviderAdministrator
  objectClass: groupOfUniqueNames
  description: e-Utility Provider Administrator
  uniquemember: cn=dexterodsadmin,dc=ibm,dc=com

dn: cn=OfferingSubscriber,dc=ibm,dc=com
  cn: OfferingSubscriber
  objectClass: groupOfUniqueNames
  description: Offering Subscriber
  uniquemember: cn=dexteremea,dc=ibm,dc=com
  uniquemember: cn=dexteramericas,dc=ibm,dc=com
```

The following example shows the definitions of the above groups using the `ldapmodify` command.

**Example 2   Output of ldapmodify command**

```plaintext
ldapmodify -i your_config_file.ldif -h db2srv -w tdsuser -D "cn=root"
modifying entry cn=OfferingProviderAdministrator,dc=ibm,dc=com
modifying entry cn=OfferingServiceManager,dc=ibm,dc=com
modifying entry cn=ODSProviderAdministrator,dc=ibm,dc=com
modifying entry cn=OfferingSubscriber,dc=ibm,dc=com
```
IBM Dynamic Infrastructure for mySAP installation on the Tivoli Provisioning Manager server

The Standard Installer for IBM Dynamic Infrastructure for mySAP was used and the instructions on the screen were fairly self-explanatory. For further information, refer to *IBM Dynamic Infrastructure Enterprise Edition for mySAP Business Suite Installation and Customization, BOEE-EUIN*.

Before you begin the installation procedure, make sure the following actions have been performed:

- The management environment meets the hardware and software prerequisites.
- Java Runtime Environment 1.4.1 or later is installed on the management server.
- All mandatory patches have been applied to IBM Tivoli Provisioning Manager V2.1 and IBM WebSphere Application Server. For the latest patches and fixes, visit the Tivoli Provisioning Manager Web site at: 
  

- Ensure that the SAP Java Connector (JCo) is installed on the management server. SAP JCo It is not included in the regular SAP customer release. The current JCo package is available at the SAP Service Marketplace Web page at http://service.sap.com/connectors. In order to download the JCo package, you must have an SAP Service Marketplace account.

- The latest version of the IBM Dynamic Infrastructure for mySAPQ–supported managed-through (boot servers) servers should be installed on the Tivoli Provisioning Manager server. These are the NIM or z/VM automation packages. The latest Tivoli Provisioning Manager automation packages are available at the IBM Orchestration and Provisioning Automation Library: 
  

In our case study scenario, we must have the latest version of the z/VM automation package, as shown in Figure 3 on page 13.

**Note:** When creating groups in the LDIF file, you must create their Distinguished Name (DN) with the base DN used for the Tivoli Provisioning Manager (dc=ibm, dc=com).

For example, dn: cn=OfferingSubscriber,dc=ibm,dc=com.
We have already configured LDAP security, so we can select **Skip LDAP Configuration** during IBM Dynamic Infrastructure for mySAP installation. See “Configuring LDAP security” on page 14 for details about this configuration.

**Verify installation of IBM Dynamic Infrastructure for mySAP components**

IBM Dynamic Infrastructure for mySAP includes middleware provisioning and application provisioning as part of the offering. This is accomplished through the IBM Dynamic Infrastructure for mySAP automation package, which is shipped with the IBM Dynamic Infrastructure for mySAP installation media and is available in the euisap.tcdriver file.

The automation package introduces new Device Models, logical operations, workflows, Java plug-ins, scripts, and configuration templates. Refer to the automation package documentation for a complete list and details.

The automation package is installed automatically by the IBM Dynamic Infrastructure for mySAP installation procedure.

For all errors during the installation of the automation package, refer to the following log files:

- Installer error log
  - `<IDI_INSTALL>/logs/tcdriver-install-euisap-stderr.log`
- Installation log
  - `<IDI_INSTALL>/logs/tcdriver-install-euispa-stdout.log`
- General and additional error messages
  - `<IBM_COMMON_LOG>/COP/logs/tcdrivermanager.log`
By default, debug messages are not copied to the log files. For more log file information, change the settings on %TIO_HOME%/config/log4j.prop to:

- \log4j.appender.errorfile.threshold=debug
- \log4j.appender.consolefile.threshold=debug

To enable more driver-specific log file information, append the information in Example 3 to the %TIO_HOME%/config/log4j.prop file.

**Example 3  log4j.prop additional setting**

```properties
# CONFIGURATION OF LOGGERS FOR EUI TRACES
output.euiRasPattern=[%d{MM/dd/yy HH:mm:ss:SSS zzz}] %m%n

log4j.category.com.ibm.eui=ALL, eui_trace, consolefile
log4j.additivity.com.ibm.eui=false

# Messages and traces generated by euiRas go to eui_trace.log
# and to console.log
# Rolling after file is 10 MB.
#
log4j.appender.eui_trace=org.apache.log4j.RollingFileAppender
log4j.appender.eui_trace.MaxFileSize=10MB
log4j.appender.eui_trace.MaxBackupIndex=5
log4j.appender.eui_trace.File=${kanaha.logs}/eui_trace.log
log4j.appender.eui_trace.layout=org.apache.log4j.PatternLayout
log4j.appender.eui_trace.layout.ConversionPattern=${output.euiRasPattern}
log4j.appender.eui_trace.append=true

# Remove comment in following lines to stop the logging of messages in
# the eui_trace file
# log4j.category.com.ibm.eui.ras.EUIMessages=ALL
# log4j.additivity.com.ibm.eui.ras.EUIMessages=false
```

To verify the status of the IBM Dynamic Infrastructure for mySAP automation package in Tivoli Provisioning Manager, use the **tc-driver-manager** command.

**Example 4  euisap.tcdriver status**

```
C:\IBM\tivoli\thinkcontrol\tools>tc-driver-manager.cmd getDriverStatus euisap
TCDriverManager was started.
Config directory = 'file:C:\IBM\tivoli\thinkcontrol\config/'.
Driver directory = 'C:\IBM\tivoli\thinkcontrol\drivers/'.
ObjectView Persistency Layer version 6.5.16 looking for initialization file:
C:\
IBM/tivoli/thinkcontrol/config/ObjectView.properties
COM.ibm.db2.jdbc.app.DB2Driver version: 8.1 loaded.
```
Tivoli Provisioning Manager Data Center modeling

The Data Center Model (DCM) is a description of all logical and physical resources that are managed by Tivoli Provisioning Manager. Managed resources could be servers, switches, load balancers, software, and other related equipment. The DCM keeps track of changes made by workflows to the hardware and software configuration and keeps the model in sync with the real world hardware and software assets that are associated with it.

The DCM also stores information needed for the management of resource pools and clusters such as server IDs, size of resource pools, number of active and inactive servers, and server priorities.

The DCM information is stored in a central database controlled by Tivoli Provisioning Manager. The DCM contains information regarding a real-life data center. Tivoli Provisioning Manager communicates directly with all components of the data center based on the definitions in the DCM.

DCM entries can be built using an XML file and imported into the DCM database. To customize the our DCM XML file, we have to provide the following solution specific information:

- Switch fabric
- VLANs and subnets
- Boot server
- Software product
- Software stack
- Service Access Points
- The The Tivoli Provisioning Manager server itself
- Resource Pool and servers
- Customer, Application, and Cluster

The next sections provide details of the DCM definitions used during the development of this document.
Note: As described in “Existing IT infrastructure” on page 7, the topology chosen for this paper is topology 3, which involves the definitions of elements running on zSeries. The elements defined in the DCM presented in this section contain attributes that are specific to this topology. Refer to IBM Dynamic Infrastructure Enterprise Edition for mySAP Business Suite Installation and Customization, BOEE-EUIN-00, for details about specific attributes used by other topologies.

Switch fabric
The switch fabric is used to create a logical hierarchy of resources in the DCM. Example 5 shows the definition we used.

Example 5 Switch Fabric

```xml
<!-- SWITCH FABRIC -->
<switch-fabric name="ITSO_zVM_Fabric"/>
```

VLANs and subnets
Define here the subnetworks you are using in your setup. You may obtain this information from your LAN administrator.

Example 6 VLANs and subnets

```xml
<subnetwork ipaddress="9.12.4.0" netmask="255.255.254.0">
  <!-- zVM admin VLAN zVM system 1 -->
  <vlan vlan-number="110" fabric="ITSO_zVM_Fabric"/>
</subnetwork>
```

Boot server
This part of the XML file describes the parameters needed to access the resources of a zSeries Z/VM system. ZVM Disk Pool Name is a name for a set of minidisks used for the cloning process. The ZVM System parameter defines the IP address of the z/VM system that receives commands from the Tivoli Provisioning Manager.

Password credentials needed for the logon into the z/VM are provided by the Service Access Points parameters. These credentials are required for the provisioning operations to access the z/VM system using the VM Systems management API (VMAPI). The user ID and password information provided in these Service Access Points are of an administrative user ID defined in z/VM. Refer to “z/VM preparation and configuration” on page 33 for details.
Example 7  zVM Boot Server

```xml
<boot-server name="zVM Boot Server" is-device-model="zVM Boot Server"
locale="en_US">
  <property name="ZVM Disk Pool Name" value="sap1f1"/>
  <property name="ZVM System" value="9.12.4.129"/>
  <sap name="zVM vmapi" port="0" host="false" is-device-model="zVM Vmapi Service Access Point"
protocol-type="other" app-protocol="LOCAL-EXEC" other-type-description="execute vmapi local command" locale="en_US">
    <default-sap operation-type="execute-command"/>
    <credentials search-key="zVM login" is-default="true">
      <password-credentials username="idadm" password="newpw"/>
    </credentials>
    <credentials search-key="zVM initial" is-default="false">
      <password-credentials username="idadm" password="newpw"/>
    </credentials>
  </sap>
</boot-server>
```

Software product
This definition is used to reference an operating system to be installed on the provisioned servers.

Example 8  Software product

```xml
<software name="zVM SuSE SLES Operating System" is-device-model="zVM SuSE Operating System" version="1.0" type="OPERATING_SYSTEM" package_path="" locale="en_US">
  <property name="ZVM PingTimeoutInSeconds" component="DEPLOYMENT_ENGINE" value="3600"/>
</software>
```

Software stack
This represents the Linux on zSeries master installation to be used during the provisioning operation. The software stack definition has boot server dependencies.

Four variables are defined by the property tag in the XML definition required for IDI. They are:

- ZVM Prototype
  This variable contains the Prototype definition name as defined in z/VM.

- ZVM Master Image
  This variable contains the user ID in z/VM of the configuration master server and the set of minidisks to be used during the cloning operation (1cxx in our case).
- **ZVM CloneOperationTimeoutInSeconds**
  
The overall time required to execute the clone image operation.

- **ZVM VmapiOperationTimeoutInSeconds**
  
The overall time required to execute the VMAPI operation within the image install workflows.

**Example 9  Software stack**

```xml
<software-stack name="zVM SW Stack OS SuSE" is-device-model="zVM Image Software Stack" is-image="true" boot-server="zVM Boot Server" locale="en_US">
  <software-stack-product product-name="zVM SuSE SLES Operating System" position="1" expected-state="running"/>
  <property name="ZVM Prototype" component="DEPLOYMENT_ENGINE" value="LINUXVM"/>
  <property name="ZVM Master Image" component="DEPLOYMENT_ENGINE" value="sapcm 1"/>
  <property name="ZVM CloneOperationTimeoutInSeconds" component="DEPLOYMENT_ENGINE" value="18000"/>
  <property name="ZVM VmapiOperationTimeoutInSeconds" component="DEPLOYMENT_ENGINE" value="3600"/>
  <property name="destinationDeviceModel" component="DEPLOYMENT_ENGINE" value="zVM SuSE Operating System"/>
  <sap name="zVM ssh" port="22" host="true" is-device-model="zVM SSH Service Access Point" protocol-type="ipv4" app-protocol="SSH" auth-compulsory="false" locale="en_US">
    <default-sap operation-type="execute-command"/>
    <credentials search-key="zVM ssh sap" is-default="true">
      <rsa-credentials username="root"/>
    </credentials>
  </sap>
  <sap name="zVM scp" port="22" host="true" protocol-type="ipv4" app-protocol="SCP" auth-compulsory="false" locale="en_US">
    <default-sap operation-type="file-transfer"/>
    <credentials search-key="scp" is-default="true">
      <rsa-credentials username="root"/>
    </credentials>
  </sap>
  <sap name="zVM ping" port="0" host="true" protocol-type="ipv4" app-protocol="SNMP" auth-compulsory="false" locale="en_US">
    <default-sap operation-type="ping"/>
    <credentials search-key="default" is-default="true">
      <snmp-credentials community="public"/>
    </credentials>
  </sap>
</software-stack>
```
Service Access Points

These Service Access Point definitions are used for access authorization to the provisioned server during the provisioning operations. For example, during the cloning operation, the newly provisioned server reboots. The provisioning workflows execute an SNMP ping operation to find out whether the server is back up and running. The SNMP ping operation will fail without the Service Access Point definition snmp-get.

Example 10  Service Access Points

```xml
<!ENTITY linux-saps '>
  <sap name="snmp-set" port="161" is-device-model="SNMP V1 Service Access Point" context="write" protocol-type="ipv4" app-protocol="SNMP" locale="en_US">
    <default-sap operation-type="set-attribute"/>
    <credentials search-key="primary" is-default="true">
      <snmp-credentials community="public"/>
    </credentials>
  </sap>
  <sap name="snmp-get" port="161" is-device-model="SNMP V1 Service Access Point" context="read" protocol-type="ipv4" app-protocol="SNMP" locale="en_US">
    <default-sap operation-type="get-attribute"/>
    <credentials search-key="primary" is-default="true">
      <snmp-credentials community="public"/>
    </credentials>
  </sap>
  <sap name="ssh service" port="22" app-protocol="SSH" is-device-model="SSH Service Access Point" host="true" locale="en_US">
    <default-sap operation-type="execute-command"/>
    <credentials search-key="primary" is-default="true">
      <password-credentials username="root" password="itso4you"/>
    </credentials>
  </sap>
  <sap name="scp client" port="22" app-protocol="SCP" is-device-model="SSH Service Access Point" host="false" locale="en_US">
    <default-sap operation-type="file-transfer"/>
    <credentials search-key="primary" is-default="true">
      <password-credentials username="root" password="itso4you"/>
    </credentials>
  </sap>
'>

The Tivoli Provisioning Manager server

This DCM definition represents the server that is running Tivoli Provisioning Manager. This definition will mainly be used as an SSH client for SSH communications to other resources in the DCM.
Example 11  
Tivoli Provisioning Manager server DCM entry

```xml
<server name="tpmsrv.itso.ibm.com" locale="en_US">
  <sap name="zVM ssh" port="0" host="false" is-device-model="zVM SSH Service Access Point" protocol-type="ipv4" app-protocol="SSH" auth-compulsory="false">
    <default-sap operation-type="execute-command"/>
    <credentials search-key="zVM ssh sap" is-default="true">
      <rsa-credentials username="root"></rsa-credentials>
    </credentials>
  </sap>
  <sap name="zVM scp" port="22" host="false" protocol-type="ipv4" app-protocol="SCP" auth-compulsory="false">
    <default-sap operation-type="file-transfer"/>
    <credentials search-key="scp" is-default="true">
      <rsa-credentials username="root"></rsa-credentials>
    </credentials>
  </sap>
</server>
```

Resource Pool and servers

The Resource Pool contains a series of predefined servers that can be dynamically allocated to applications using the application's Cluster definition (defined below). During server allocation time, a server defined in the Resource Pool is associated with a z/VM user ID (virtual machine) that will be installed the previously defined software stack.

A variable defined by the property tag in the XML definition is required for IDI. The variable name must be `zVM Related BootServerId` and its value must contain the ID defined in Tivoli Provisioning Manager for the boot server. We defined the value of this variable as `X` because the boot server had not been defined in the DCM yet. After all definitions were imported into the DCM, we used the Tivoli Provisioning Manager GUI to manually change the value of this variable with the correct Boot Server ID.

Note that the device model defined for the servers is `euisap zVM Server`. This device model is defined by the IBM Dynamic Infrastructure for mySAP automation package. Because this automation package is installed in Tivoli Provisioning Manager during the IBM Dynamic Infrastructure for mySAP installation, ensure that the Resource Pool and servers definitions are done after the IBM Dynamic Infrastructure for mySAP installation.
Example 12 Resource Pools and servers

```xml
<spar-pool name="zVM Virtual Server Pool" vlan="110" os-type="linux"
fabric="ITSO_zVM_Fabric" locale="en_US">
    <!-- add servers as representation for the virtual servers capacity to
the spare-pool with name="zVM Server Pool" -->
    <server name="Server 1" is-device-model="euisap zVM Server"
locale="en_US">
        <network-interface ipaddress="9.12.4.250" name="Admin"
netmask="255.255.254.0" managed="false"/>
        <nic managed="true" connected-to-switch="Prod-SW-1"
connected-to-module="fa0" connected-to-port="11"/>
        <property name="zVM Related BootServerId"
component="DEPLOYMENT_ENGINE" value="X"/>
        &linux-saps;
    </server>
    <server name="Server 2" is-device-model="euisap zVM Server"
locale="en_US">
        <network-interface ipaddress="9.12.4.249" name="Admin"
netmask="255.255.254.0" managed="false"/>
        <nic managed="true" connected-to-switch="Prod-SW-1"
connected-to-module="fa0" connected-to-port="12"/>
        <property name="zVM Related BootServerId"
component="DEPLOYMENT_ENGINE" value="X"/>
        &linux-saps;
    </server>
    <server name="Server 3" is-device-model="euisap zVM Server"
locale="en_US">
        <network-interface ipaddress="9.12.4.248" name="Admin"
netmask="255.255.254.0" managed="false"/>
        <property name="zVM Related BootServerId"
component="DEPLOYMENT_ENGINE" value="X"/>
        &linux-saps;
    </server>
    <server name="Server 4" is-device-model="euisap zVM Server"
locale="en_US">
        <network-interface ipaddress="9.12.4.247" name="Admin"
netmask="255.255.254.0" managed="false"/>
        <property name="zVM Related BootServerId"
component="DEPLOYMENT_ENGINE" value="X"/>
        &linux-saps;
    </server>
</spar-pool>
```
**Customer, Application, and Cluster**

This section describes a sample customer who uses an application defined in the DCM.

When a customer makes a subscription to an offering, the workflows triggered in Tivoli Provisioning Manager dynamically define DCM entries for Customer, Application, and Cluster related to the subscription. As these operations occur dynamically, there is no need to define them manually. However, it is a good idea to make the definitions for testing and troubleshooting purposes. Example 13 presents a sample DCM entry for Customer, Application, and Cluster.

**Example 13  Customer, Application and Cluster sample definition**

```xml
<!-- this is a sample customer for manual provisioning and test purposes -->
<customer name="zVM Customer">
  <application name="zVM App1" priority="1" locale="en_US">
    <service-level-objective internal="false" type="min-availability" value="0.99"/>
    <service-level-objective internal="false" type="max-response-time" value="0.167"/>
    <cluster name="zVM Cluster App1" is-device-model="zVM Cluster" min-servers="0" max-servers="12" pool="zVM Virtual Server Pool" fabric="ITSO_zVM_Fabric" vlan="110" software-stack="zVM SW Stack OS SuSE" locale="en_US"/>
  </application>

  <application name="zVM App2" priority="1" locale="en_US">
    <service-level-objective internal="false" type="min-availability" value="0.99"/>
    <service-level-objective internal="false" type="max-response-time" value="0.167"/>
    <cluster name="zVM Cluster App2" is-device-model="zVM Cluster" min-servers="0" max-servers="12" pool="zVM Virtual Server Pool" fabric="ITSO_zVM_Fabric" vlan="110" software-stack="zVM SW Stack OS SuSE" locale="en_US"/>
  </application>

  <!-- add additional application for this customer here -->
</customer>
```
IBM Dynamic Infrastructure for mySAP configuration

Configuration and customization of the IBM Dynamic Infrastructure for mySAP is done via three files:

- Offering0001.xml
- SubscriptionConfiguration.xml
- basicConfiguration.properties

The entries in these files also directly relate to entries in the DCM mentioned earlier. Additional information is available in *IBM Dynamic Infrastructure Enterprise Edition for mySAP Business Suite Installation and Customization*, BOEE-EUIN-00.

Offering0001.xml

Example 14 on page 28 is the Offering0001.xml file that we modified from the template, as per instructions in *IBM Dynamic Infrastructure Enterprise Edition for mySAP Business Suite Installation and Customization*, BOEE-EUIN. We modified the following parameters:

- tpmUserid
  Tivoli Provisioning Manager administrator user ID. We used tioappadmin.

- tpmPwd
  Tivoli Provisioning Manager administrator password.

- variationId
  The topology that you want IBM Dynamic Infrastructure for mySAP to support for your installation. In our case, we chose variationId030.

- variationName
  This presents a choice of topologies to subscribers. In our case study environment, we use only AppServer: zVM/ zLinux; Database: DB2/zSeries.

- createWfAttribute key=poolId
  This is the ID of the resource pool defined for the Application cluster in Tivoli Provisioning Manager. This ID is available after importing the DCM configurations presented in “Tivoli Provisioning Manager Data Center modeling” on page 19. In our environment this is the ID of the zVM Virtual Server Pool resource pool.

- createWfAttribute key=switchFabricId
  This is the ID of the switch fabric defined in Tivoli Provisioning Manager. In our environment this is the ID of the ITS0_zVM_Fabric switch fabric.
createWfAttribute key=meteringMessageJMSUserId
Tivoli Provisioning Manager internal administrator user ID. We used tiointernal.

createWfAttribute key=meteringMessageJMSPwd
Tivoli Provisioning Manager internal administrator user ID password.

createWfAttribute key=errorMessageJMSUserId
Tivoli Provisioning Manager internal administrator user ID. We used tiointernal.

createWfAttribute key=errorMessageJMSPwd
Tivoli Provisioning Manager internal administrator user ID password.

Example 14 offering0001.xml file

```xml
<?xml version="1.0" encoding="UTF-8"?>
<offering xmlns="http://www.ibm.com/xmlns/prod/DynamicInfrastructure"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
../OfferingCantata.xsd">
  <offeringId key="offeringId" value="Offering0001" reference="none" type="string"/>
  <offeringName key="offeringName" value="IBM Dynamic Infrastructure for mySAP Version 1.0" reference="CP" type="string"/>
  <provider>
    <providerId key="providerId" value="providerName" reference="CP" type="string"/>
    <orderPreparationUrl key="orderPreparationUrl" value="http://localhost:9080/euiSapGuiWeb/wsdl/SapSubscriptionControllerPort.wsdl" reference="SE" type="string"/>
    <seServiceUrl key="seServiceUrl" value="http://localhost:9080/euiSeWeb/wsdl/com/ibm/eui/se/ServiceEnvironmentService.wsdl" reference="SE" type="string"/>
    <tpmUrl key="tpmUrl" value="http://localhost:9080/tcSoap/servlet/rpcrouter" reference="SE" type="string"/>
    <tpmUserId key="tpmUserId" value="tioappadmin" reference="SE" type="string"/>
    <tpmPwd key="tpmPwd" value="tioappadmin" reference="SE" type="string"/>
    <providerAdminLanguage key="providerAdminLanguage" value="EN" reference="CA" type="string"/>
  </provider>
  <variation>
    <variationId key="variationId" value="varId030" reference="none" type="string"/>
  </variation>
</offering>
```
<variationName key="variationName" value="AppServer: zVM/ zLinux; Database: DB/ zSeries" reference="CP" type="string"/>
<linkToOptionsAndWfs>
<linkToCreateWF>
<createServerWfId key="createWfId" value="euisap.CreateServiceEnvironment" reference="SE" type="string"/>
<createWfAttribute key="createServerWf" value="euisap.ClusterAddServer" reference="SE" type="string"/>
<createWfAttribute key="applicationName" value="AppServer: zLinux/zVM; Database: DB2/ zSeries" reference="SE" type="string"/>
<createWfAttribute key="clusterName" value="zSeries SAP2" reference="SE" type="string"/>
<createWfAttribute key="poolId" value="1320" reference="SE" type="string"/>
<createWfAttribute key="switchFabricId" value="1242" reference="SE" type="string"/>
<createWfAttribute key="errorMessageJMSHost" value="localhost" reference="SE" type="string"/>
<createWfAttribute key="errorMessageJMSPort" value="2809" reference="SE" type="string"/>
<createWfAttribute key="errorMessageJMSUserId" value="tiointernal" reference="SE" type="string"/>
<createWfAttribute key="errorMessageJMSPwd" value="internal" reference="SE" type="string"/>
</linkToCreateWF>
<linkToDeleteWF>
<deleteServerWfId key="deleteWfId" value="euisap.DeleteServiceEnvironment" reference="SE" type="string"/>
<deleteWfAttribute key="deleteServerWf" value="euisap.ClusterRemoveServer" reference="SE" type="string"/>
<deleteWfAttribute key="cleanupWf" value="euisap.CleanServiceEnvironment" reference="SE" type="string"/>
<deleteWfAttribute key="cleanupServerWf" value="euisap.ClusterCleanupServer" reference="SE" type="string"/>
</linkToDeleteWF>
</linkToOptionsAndWfs>
SubscriptionConfiguration.xml

Example 15 on page 31 is the SubscriptionConfiguration.xml file that we modified from the template based on instructions in *IBM Dynamic Infrastructure Enterprise Edition for mySAP Business Suite Installation and Customization*, BOEE-EUIN. We modified the following parameters:

- **sapSystemId**
  The SAP system ID of the base SAP system. In our case, m5d.

- **centralInstance**
  Short host name of the SAP Central Instance: sapci.

- **sapSystemNumber**
  System number of the base SAP system. In our environment 00.

- **sapClient**
  The SAP client number. In our environment 000.

- **sapUserId**
  SAP user ID with DDIC access rights for SAP agent component of metric service for SAP system logon. In our environment m5dopt.

- **sapPassword**
  The password of sapUserId.

- **sapLogonGroup**
  The SAP logon group for provisioned SAP application servers. In our environment IDIGRP.

- **softwareStackId**
  This is the software stack ID defined during the Tivoli Provisioning Manager data center model configuration.

- **customerVlanId**
  This is the VLAN ID defined during the Tivoli Provisioning Manager data center model configuration.

- **customerVlanName**
  This is the name of the VLAN defined during the Tivoli Provisioning Manager data center model configuration. In our environment it is ITSO_zVM_Fabric - 110.
- **databaseVlanId**
  This is the VLAN ID in which the database server hosting the IBM Dynamic Infrastructure for mySAP database resides. This value is defined during the Tivoli Provisioning Manager data center model configuration.

- **databaseVlanName**
  This is the VLAN ID in which the database server hosting the IBM Dynamic Infrastructure for mySAP database resides. This value is defined during the Tivoli Provisioning Manager data center model configuration. In our environment it is **ITSO_zVM_Fabric - 110**.

- **ldomainSuffix**
  The domain suffix defined in the LDAP configuration. Refer to “Configuring LDAP security” on page 14. In our environment it is **itso.ibm.com**.

**Example 15  SubscriptionConfiguration.xml file**

```xml
<configurationParameter xmlns="http://www.ibm.com/xmlns/prod/DynamicInfrastructure"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
../ConfigurationParameter.xsd">
  <configAttribute key="sapSystemId" value="M5D" reference="SE" type="string"/>
  <configAttribute key="centralInstance" value="sapci" reference="CA" type="string"/>
  <configAttribute key="dlProvMinCpuLoad" value="20" reference="SE" type="string"/>
  <configAttribute key="dlProvFinishedTime" value="1500000" reference="SE" type="string"/>
  <configAttribute key="dlSampleInterval" value="60000" reference="SE" type="string"/>
  <configAttribute key="dlDeprovMaxNumConnectedUsers" value="3" reference="SE" type="string"/>
  <configAttribute key="sapClient" value="000" reference="SE" type="string"/>
  <configAttribute key="sapLogonGroup" value="idigrp" reference="SE" type="string"/>
  <configAttribute key="sapPassword" value="itso4you" reference="SE" type="string"/>
  <configAttribute key="sapRouterString" value="" reference="SE" type="string"/>
  <configAttribute key="sapSystemNumber" value="00" reference="SE" type="string"/>
  <configAttribute key="sapUserId" value="m5dopt" reference="SE" type="string"/>
  <configAttribute key="sloMaxResponseTime" value="10000" reference="SE" type="string"/>
  <configAttribute key="sapAgentRfcTimeout" value="30000" reference="SE" type="string"/>
  <configAttribute key="msSapMessageTextDeprovisioningRow1" value="WARNING: Your server will shut down soon." reference="SE" type="string"/>
  <configAttribute key="msSapMessageTextDeprovisioningRow2" value="Please re-login to the SAP system in order to be assigned" reference="SE" type="string"/>
  <configAttribute key="msSapMessageTextDeprovisioningRow3" value="to another application server." reference="SE" type="string"/>
  <configAttribute key="msSapMessageTextProvisioningRow1" value="INFO-Message: New servers have been provisioned." reference="SE" type="string"/>
  <configAttribute key="msSapMessageTextProvisioningRow2" value="Your current server runs a high work load. If you" reference="SE" type="string"/>
</configurationParameter>
```
<configAttribute key="msSapMessageTextProvisioningRow3" value="re-login you are assigned to a faster server." reference="SE" type="string"/>
<configAttribute key="msSapMessageTextProvisioningSendFlag" value="true" reference="SE" type="string"/>
<configAttribute key="softwareStackId" value="1248" reference="SE" type="string"/>
<configAttribute key="customerVlanId" value="1244" reference="SE" type="string"/>
<configAttribute key="customerVlanName" value="ITSO_zVM_Fabric – 110" reference="SE" type="string"/>
<configAttribute key="databaseVlanId" value="1244" reference="SE" type="string"/>
<configAttribute key="databaseVlanName" value="ITSO_zVM_Fabric – 110" reference="SE" type="string"/>
<configAttribute key="maxNumberOfServers" value="4" reference="SE" type="string"/>
<configAttribute key="minNumberOfServers" value="1" reference="SE" type="string"/>
<configAttribute key="domainSuffix" value="itso.ibm.com" reference="SE" type="string"/>
</configurationParameter>

**basicConfiguration.properties**

The basicConfiguration.properties file allows for customization of many parameters for IBM Dynamic Infrastructure for mySAP. We have changed the parameters as highlighted in Example 16.

*Example 16  basicConfiguration.properties file*

```
# context factory - should not be changed
domain.objcontext=com.ibm.websphere.naming.WsnInitialContextFactory
# Host providing jndi naming service for message logging component
java.naming.provider.url=iiop://localhost:2809/

# jms uid & pw
jms.eui.userid=tiointernal
jms.eui.password=internal

# host running messageservice

# host running ubs

# host running service environment

# host running optimizer
```
# host running message sender

# port used for meterevent xfer between GUI & ubs host
socket_connection_port=44408

# Number of messages held in wraparound buffer in storage by message logging component
com.ibm.eui.log.message_count=300

# name of configuration directory below IDI_HOME
eui.configuration.root=configuration

# host running the optimizer

# metric service reconnect interval to optimizer in ms
com.ibm.eui.ms.connectToOptimizerWaitBeforeNextRetry=3000

# jms definitions for metric service
com.ibm.eui.ms.optimizerJMSQueues.hostname=localhost
com.ibm.eui.ms.optimizerJMSQueues.port=2809
com.ibm.eui.ms.optimizerJMSQueues.userid=tiointernal
com.ibm.eui.ms.optimizerJMSQueues.password=internal

---

**z/VM preparation and configuration**

Basic steps of associated workflows for the provisioning operations in our case study scenario are performed using a command line interface to the z/VM system management API (VMAPI) with extensions to handle Linux on zSeries system images and clone these from a manually prepared master system into automatically created clone systems.

Several tasks must be performed for the provisioning operation described above to work properly. This might involve tasks such as preconfiguring the target system by providing preconfigured operating system images or establishing proper authorization for basic operations. The tasks performed for our case study scenario are described in this section. Refer to the z/VM product manuals at [http://www.vm.ibm.com/](http://www.vm.ibm.com/) for more information about performing these tasks.
Linux directory prototype definitions

To support the creation of virtual machines and the application of images into these virtual machines, a special customization is required for DirMaint™ (DIRM) using native z/VM. We defined the default Linux profile used in each of the z/VM Guest user definitions. This is where we were able to specify the default building blocks to be used by IDI workflows for the creation of the user IDs associated with each dynamically provisioned server.

The following example shows the Linux prototype directory entry used in this scenario. In our case, the NICDEF statement refers to the virtual network device address of C200, which can be used by each dynamically provisioned server without parameter file modification and allows connectivity via the z/VM vswitch definition (presented later).

Example 17  Linux prototype directory entry

```
USER LINUXVM NOLOG
  INCLUDE LINDFLT
  NICDEF C200 TYPE QDIO LAN SYSTEM LAN1 MACID 010000
  MDISK 191 3390 AUTOG 0010 ANY MR
```

The following example shows the Linux default profile that is referenced during virtual machine creation of our provisioned servers.

Example 18  Linux default profile

```
PROFILE LINDFLT
  CLASS G
  STORAGE 512M
  MAXSTORAGE 2047M
  IPL C02
  IUCV ALLOW
  MACHINE ESA
  CONSOLE 0009 3215 T
  SPOOL 000C 2540 READER *
  SPOOL 000D 2540 PUNCH A
  SPOOL 000E 1403 A
  LINK MAINT 0190 0190 RR
  LINK MAINT 019D 019D RR
  LINK MAINT 019E 019E RR
  *DVHOPT LNK0 LOG1 RCM1 SMS0 NPW1 LNGAMENG PWC20050413 CRCêd
```
Administrative user ID definition

The following example shows our definition of the administrative user ID in z/VM that will be used when Tivoli Provisioning Manager uses the VM VSM API to create new directory entries for the newly provisioned servers. We created the IDADM guest as outlined in the IDI for mySAP Business Suite Installation and Customization. This guest is for providing administrative services to the IBM DI workflows that are run on the Tivoli management servers.

Example 19  Administrative user ID

```
USER IDADM XXXXXXXX
  INCLUDE IBMDFLT
    MDISK 0191 3390 1081 20 515U1R MR
  *DVHOPT LNK0 LOG1 RCM1 SMS0 NPW1 LNGAMENG PWC20050413 CRCh6
```

The administrative user ID must be authorized to perform tasks encompassed by DirMaint command level 150A and command sets ADGHP. Under certain circumstances command level 140A is needed. For example, to authorize the VM user ID IDADM to perform the DirMaint task needed for dynamic provisioning, the following DirMaint authorization command must be issued by an appropriately authorized VM user:

```
DIRM FOR ALL AUTHFOR IDADM CMDLEVEL 150A CMDSET ADGHP
```

Example 20 shows a sample entry created in z/VM for a new application server during the dynamic provisioning operation. This entry is created in the z/VM directory based on the Linux prototype, SAP configuration master server, and the Administrative user ID definitions.

Example 20  User ID definition for a cloned system

```
USER IDADM002 XXXXXXXX
  INCLUDE LINDFLT
  IPL C02
    NICDEF C200 TYPE QDIO LAN SYSTEM LAN1 MACID 010002
    MDISK 0191 3390 1141 10 515U1R MR
    MDISK 0C01 3390 001 3338 DK8024 MR
    MDISK 0C02 3390 001 400 DK8025 MR
    MDISK 0C03 3390 001 3338 DK8026 MR
    MDISK 0C04 3390 001 3338 DK8027 MR
  *DVHOPT LNK0 LOG1 RCM1 SMS0 NPW1 LNGAMENG PWC20050720 CRCh6
```
VM Systems Management Server settings

The VM Systems Management Server must be defined, configured, and active at all times for the dynamic provisioning operations. The authorization control files of the VM Systems Management Server must be customized.

Example 21 shows our VM Systems Management Server directory entry.

Example 21  VSMSERVE directory definition

```
USER VSMSERVE XXXXXXXX 32M 32M ABCDEFG
  DBONECMD FAIL
  IPL CMS
  IUCV ANY MSGLIMIT 255
  MACHINE ESA
  OPTION DIAG88 MAINTCCW LNKS LNKE
  CONSOLE 0009 3215 T
  SPOOL 000C 2540 READER *
  SPOOL 000D 2540 PUNCH A
  SPOOL 000E 1403 A
  LINK MAINT 0190 0190 RR
  LINK MAINT 019D 019D RR
  LINK MAINT 019E 019E RR
  LINK MAINT 0193 0193 RR
  LINK MAINT 0CF1 0CF1 MD
  LINK MAINT 0CF2 0CF2 MD
  LINK TCPMAINT 0591 0591 RR
  LINK TCPMAINT 0592 0592 RR
  MDISK 0191 3390 1818 025 515RES MR READ WRITE MULTIPLE
```

Authorization control files for the VM Systems Management API must be customized to grant the administrative user (IDADM in our case) proper authorization access to the VMAPI functions. These authorization control files reside on the 191 minidisk of the VSMSERVE user ID and are named VMSERVE AUTHLIST and VMSERVE NAMELIST. The administrative user ID must to be authorized to perform the following group of functions for all user IDs:

```
QUERY_FUNCTIONS
```

It also must be authorized to perform this group of functions for all user IDs defined to be used during the provisioning operation (IDADM00X user IDs in our case):

- DM_CONTROL
- DM_IMAGE_CONTROL
- IMAGE_OPERATION
- IMAGE_DEVICES
- IMAGE_CONNECT
To simplify our setup, we granted the user ID IDADM full access (Example 22), which is not recommended for production environments.

**Example 22  VSMServe AUTHLIST configuration file**

<table>
<thead>
<tr>
<th>MAINT</th>
<th>ALL</th>
</tr>
</thead>
<tbody>
<tr>
<td>IDADM</td>
<td>ALL</td>
</tr>
<tr>
<td>VSMServe</td>
<td>ALL</td>
</tr>
</tbody>
</table>

Example 23 shows the VSMServe NAMELIST used in our scenario.

**Example 23  VSMServe NAMELIST file**

```
:nick.SERVER_FUNCTIONS :list.
 MODIFY_SESSION_TIMEOUT_INTERVAL
 MODIFY_SERVER_TIMEOUT_INTERVAL
 QUERY_TIMEOUT_INTERVAL
 LOGIN
 LOGOUT
 TRUSTED_USER
 VSMAPI_DEBUG
 VSMAPI_NULL
 VSMServe_SHUTDOWN
 :nick.AUTHORIZATION :list.
 AUTHORIZATION_LIST_ADD
 AUTHORIZATION_LIST_REMOVE
 AUTHORIZATION_LIST_QUERY
 :nick.NAME_LIST :list.
 NAME_LIST_ADD
 NAME_LIST_REMOVE
 NAME_LIST_DESTROY
 NAME_LIST_QUERY
 :nick.DM_CONTROL :list.
 STATIC_IMAGE_CHANGES_ACTIVATE
 STATIC_IMAGE_CHANGES_DEACTIVATE
 STATIC_IMAGE_CHANGES_IMMEDIATE
 DIRECTORY_MANAGER_COMMAND
 QUERY_DIRECTORY_MANAGER_LEVEL
 QUERY_ASYNCHRONOUS_OPERATION
 :nick.DM_PROTOTYPE_MANAGEMENT :list.
 PROTOTYPE_NAME_QUERY
 PROTOTYPE_CREATE
 PROTOTYPE_REPLACE
 PROTOTYPE_DELETE
```
DASD allocation group definitions

One or more allocation groups must be defined to DirMaint. These allocation
groups are referenced during the creation of clone disks for the dynamic
provisioned servers. They define the free disk space to be used when allocating
new minidisks for those servers.

The EXTENT CONTROL file on DIRMAINT 1DF minidisk has to be modified to
include DASD Groups for allocation for Linux Guests.

We also created a group allocation for definition of available DASD devices that
are available to dynamically provisioned servers. The group definition is
contained in the DIRM Extent Control File and represented by the line beginning
with SAPFL1, where SAPFL1 is used to provide full pack minidisk allocations for the dynamic provisioned servers. Example 24 shows our configuration.

**Example 24  Extent Control file entries**

```plaintext
:REGIONS.
   *RegionId  VolSer  RegStart  RegEnd  Dev-Type  Comments
  515RES      515RES     001     3338 3390-03  3390 1
  515W01      515W01     001     3338 3390-03  3390 1
  515W02      515W02     001     3338 3390-03  3390 1
  515U1R      515U1R     001     3338 3390-03  3390 1
  DK8024      DK8024     001     3338 3390-03  3390 1
  DK8025      DK8025     001     3338 3390-03  3390 1
  DK8026      DK8026     001     3338 3390-03  3390 1
  DK8027      DK8027     001     3338 3390-03  3390 1
:END.
:GROUPS.
   *GroupName RegionList
  ANY      515U1R
SAPFL1 (ALLOCATE ROTATING)
SAPFL1 DK8024 DK8025 DK8026 DK8027 DK8020 DK8021 DK8022 DK8023
:END.
:EXCLUDE.
   * UserId Address
  MAINT 012*
  SYSDUMP1 012*
:END.
:AUTOBLOCK.
   * IBM supplied defaults are contained in the AUTOBLK DATADVH file.
   * The following are customer overrides and supplements.
   *
   *DASDType BlockSize Blocks/Unit Alloc_Unit Architecture
:END.
:DEFAULTS.
   * IBM supplied defaults are contained in the DEFAULTS DATADVH file.
   * The following are customer overrides and supplements.
   *
   *DASDType Max-Size
  3390              3339
  3390-01           1113
  3390-02           2226
  3390-03           3339
  3390-09           10017
:END.
```

It is important that you turn off all MINIDISK caching for the DASD of all dynamically provisioned servers. If you do not, the chances of cloning failures and possible file system corruptions is much higher.
Because we are allowing multiple Linux on zSeries images to simultaneously access the same disk drives, we turned off CMS minidisk caching. The CMS commands we used while logged on to MAINT to turn this feature off and then display their status are:

```
SET MDC RDEV beginning_device ending_device OFF
query mdc rdev 8020-8028, 8120-8128, 8220-8228, 8236-8237
```

Example 25 shows the output display of the status of our minidisks.

```
Example 25   CP query mdc rdev output

Minidisk cache OFF for 8020
Minidisk cache OFF for 8021
Minidisk cache OFF for 8022
Minidisk cache OFF for 8023
Minidisk cache OFF for 8024
Minidisk cache OFF for 8025
Minidisk cache OFF for 8026
Minidisk cache OFF for 8027
Minidisk cache OFF for 8028
Minidisk cache DFLT OFF for 8120
Minidisk cache OFF for 8121
Minidisk cache OFF for 8122
Minidisk cache OFF for 8123
Minidisk cache OFF for 8124
Minidisk cache OFF for 8125
Minidisk cache OFF for 8126
Minidisk cache OFF for 8127
Minidisk cache OFF for 8128
Minidisk cache OFF for 8220
Minidisk cache OFF for 8221
Minidisk cache OFF for 8222
Minidisk cache OFF for 8223
Minidisk cache OFF for 8224
Minidisk cache OFF for 8225
Minidisk cache OFF for 8226
Minidisk cache OFF for 8227
Minidisk cache OFF for 8228
Minidisk cache OFF for 8236
Minidisk cache OFF for 8237
```

**z/VM switch configuration**

We created a z/VM vswitch definition with the MAINT user ID to allow our dynamically provisioned SAP application servers to be configured with virtual network device addresses and further simplify the overall network configuration.
Note: A customer implementation likely will consist of multiple vswitch definitions and could also make use of Hipersockets, additional network adapters, and VLANs.

The vswitch definition can be seen in Example 26.

Example 26  CP query vswitch detail output

<table>
<thead>
<tr>
<th>VSWITCH SYSTEM LAN1</th>
<th>Type: VSWITCH Connected: 2</th>
<th>Maxconn: INFINITE</th>
</tr>
</thead>
<tbody>
<tr>
<td>PERSISTENT RESTRICTED NONROUTER Accounting: OFF</td>
<td></td>
<td></td>
</tr>
<tr>
<td>VLAN Unaware State: Ready</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IPTTimeout: 5 QueueStorage: 8</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Portname: UNASSIGNED RDEV: 2E2C Controller: TCPIP VDEV: 2E2C</td>
<td></td>
<td></td>
</tr>
<tr>
<td>VSWITCH Connection:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RX Packets: 716996 Discarded: 2390 Errors: 0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TX Packets: 631703 Discarded: 0 Errors: 0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RX Bytes: 514223955 TX Bytes: 383407448</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Device: 2E2E Unit: 002 Role: DATA</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Adapter Owner: SAPCM NIC: C200 Name: OSA2E20</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RX Packets: 235761 Discarded: 0 Errors: 0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TX Packets: 108479 Discarded: 0 Errors: 0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RX Bytes: 234189433 TX Bytes: 14416341</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Device: C202 Unit: 002 Role: DATA</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Options: Broadcast Multicast IPv6 IPv4 VLAN</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Unicast IP Addresses:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9.12.4.98 MAC: 02-00-00-01-00-00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>FF80::200:0:101:0 MAC: 02-00-00-01-00-00 Local</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Multicast IP Addresses:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>224.0.0.1 MAC: 01-00-5E-00-00-01</td>
<td></td>
<td></td>
</tr>
<tr>
<td>FF02::1 MAC: 33-33-00-00-00-01 Local</td>
<td></td>
<td></td>
</tr>
<tr>
<td>FF02::1:FF01:0 MAC: 33-33-FF-01-00-00 Local</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Adapter Owner: SAPDI NIC: C200 Name: OSA2E20</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RX Packets: 317405 Discarded: 0 Errors: 0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TX Packets: 385458 Discarded: 0 Errors: 0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RX Bytes: 246937824 TX Bytes: 370368819</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Device: C202 Unit: 002 Role: DATA</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Options: Broadcast Multicast IPv6 IPv4 VLAN</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Unicast IP Addresses:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9.12.4.99 MAC: 02-00-00-00-99-99</td>
<td></td>
<td></td>
</tr>
<tr>
<td>FF80::200:0:100:9999 MAC: 02-00-00-00-99-99 Local</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Multicast IP Addresses:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>224.0.0.1 MAC: 01-00-5E-00-00-01</td>
<td></td>
<td></td>
</tr>
<tr>
<td>FF02::1 MAC: 33-33-00-00-00-01 Local</td>
<td></td>
<td></td>
</tr>
<tr>
<td>FF02::1:FF00:9999 MAC: 33-33-FF-00-99-99 Local</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Example 27 shows a list of authorized users of this vswitch definition.

**Note:** Because access to a vswitch definition can be authorized prior to the creation of z/VM directory entries, it is recommended that all possible dynamically provisioned servers be included in the authorization files.

### Example 27  CP query vswitch access output

<table>
<thead>
<tr>
<th>VSWITCH</th>
<th>SYSTEM LAN1</th>
<th>Type: VSWITCH</th>
<th>Connected: 2</th>
<th>Maxconn: INFINITE</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>PERSISTENT</td>
<td>RESTRICTED</td>
<td>NONROUTER</td>
<td>Accounting: OFF</td>
</tr>
<tr>
<td></td>
<td>VLAN Unaware</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>State:</td>
<td>Ready</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IPTimeout:</td>
<td>5</td>
<td>QueueStorage: 8</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Portname:</td>
<td>UNASSIGNED</td>
<td>RDEV: 2E2C</td>
<td>Controller: TCPIP</td>
<td>VDEV: 2E2C</td>
</tr>
<tr>
<td>Authorized userids:</td>
<td>IDADM001</td>
<td>IDADM002</td>
<td>IDADM003</td>
<td>IDADM004</td>
</tr>
<tr>
<td></td>
<td>SAPDI</td>
<td>SYSTEM</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

## SAP master systems setup

A base SAP system implementation is needed prior to running the IBM Dynamic Infrastructure for mySAP. This would entail the SAP Central Instance and at least one SAP Dialog Server for each SAP SID in your landscape, along with any systems required for interfaces to other applications. It is expected that these systems already exist and are functional.

After the base SAP systems are in place, you can implement the rest of the SAP configurations to support IBM Dynamic Infrastructure for mySAP.

The Linux on zSeries code was installed by initially booting from VM EXEC starter files, then continuing the Linux on zSeries code installation over the network. Service Pack 3 was then installed via YaST2 control center.

### Updates to the SAP base systems

The SAP Service Marketplace provides the list of update activities that must be performed for each specific version of SAP. For our environment, we had to install the following additional code on each base Linux system:

- `suse-sapinit.rpm`
- `saplocales-2.2.5-2`
- `Java SDK 1.3.1`
The suse-sapinit.rpm installs several SAP-related scripts/parms to assist with the proper running of SAP. After installation of the RPM, you must modify /etc/sysconfig/sapr3 for the number of systems to auto start as well as their corresponding SAPSIDs.

Implement the new changes by entering the following command as root:

`insserv -f /etc/init.d/sapr3`

The saplocales RPM provides the SAP-specific configuration changes required to run SAP on Linux for zSeries. No manual configuration work is required after successful installation of the RPM.

We then installed the Java SDK 1.3.1.8.0 on our Linux systems per the SAP documentation.

**IBM Dynamic Infrastructure for mySAP customizations for SAP**

We created an SAP logon group to hold the dynamically provisioned servers created by the IDI workflows. Our SAP logon group for these servers is called idigrp and contains the planned servers for dynamic provisioning.

We also created separate logon groups for the day-to-day running of the SAP system, and these contained the SAP central instance and base dialog instance.

<table>
<thead>
<tr>
<th>Group</th>
<th>System</th>
</tr>
</thead>
<tbody>
<tr>
<td>admin</td>
<td>sapci_M5D_00</td>
</tr>
<tr>
<td>admin</td>
<td>sapdi_M5D_00</td>
</tr>
<tr>
<td>idigrp</td>
<td>idadm001_M5D_00</td>
</tr>
<tr>
<td>idigrp</td>
<td>idadm002_M5D_00</td>
</tr>
<tr>
<td>idigrp</td>
<td>IDADM002_M5D-00</td>
</tr>
<tr>
<td>admin</td>
<td>sapci_M5D_00</td>
</tr>
</tbody>
</table>

**The SAP Central Instance (SAPCI)**

The SAPCI system was installed using standard SAP installation tools. Refer to *SAP R/3 Enterprise on UNIX: IBM DB2 Universal Database for z/OS and OS/390*, Version 1, November 2003

The SAP Central Instance is set up to run all of the SAP processes (Dialog, Update, Enqueue, Batch, Message, Gateway, and Spool). It is large enough to
accommodate all of our base SAP processing. As a backup, we can make use of the SAPDI server for additional base resources if the need arises. Your configuration may vary from this setup and may require additional resources.

Our SAPCI system NFS exports several SAP file systems except for /sapmnt/M5D/exe, as well as some of the SAP CDs that are needed to install new systems. We NFS-mounted then exported the following file systems:

- /usr/sap/M5D
- /usr/sap/trans
- /sapmnt

We configured the SNMP control file (snmpd.conf) to allow read-only access to Public and ensured that the SNMP daemon was automatically started at system boot. Example 28 shows the actual coding of the text.

*Example 28 SAPCI snmpd.conf entry*

```
rocommunity public default
```

As this SAP system was installed prior to the IBM Dynamic Infrastructure setup, we originally defined a static real network address for this system's network connection. If we had to do this over, the team would have looked into setting up a vswitch or Hipersockets for this connection.

**Disk drive layout for SAPCI**

The SAP Central Instance was installed as a normal SAP installation. The VM user direct entry depicted in Example 29 on page 46 shows that we defined one VM minidisk for the guest 191 A disk, and 12 disk drives to hold the Linux file system, the SAP file system, and the assorted CD-ROMs that must be mounted for SAP install work to proceed.

The first Linux on zSeries disk is 0201, and it is assigned as the swap partition. Disk 0202 is assigned to /boot. Disks 0203 and 0204 are assigned to /root. This makes up the base Linux file systems needed to start Linux.

Disks 0205 through 0208, which are assigned to /sapcdds, hold the various SAP installation CD-ROMs. Disk 0209 is assigned to /install and will be the primary SAP installation directory for status and logs. Disk 0210 is assigned to the /usr/sap/trans file system, disk 0211 is assigned to /usr/sapM5D, and disk 0212 is assigned to /sapmnt: these are the SAP file systems that are accessed by all SAP systems in the landscape.
Example 29  SAPCI directory entry

```
USER SAPCI XXXXXXXX 512M 1024M G
PROFILE IBMDFLT
  SPOOL 000C 2540 READER *
  SPOOL 000D 2540 PUNCH A
  SPOOL 000E 1403 A
  CONSOLE 009 3215 T
  LINK MAINT 0190 0190 RR
  LINK MAINT 019D 019D RR
  LINK MAINT 019E 019E RR
  LINK MAINT 0402 0402 RR
  LINK MAINT 0401 0401 RR
  LINK MAINT 0405 0405 RR
  IPL CMS
  MACHINE ESA
  DEDICATE 2E24 2E24
  DEDICATE 2E25 2E25
  DEDICATE 2E26 2E26
  * 0201 is swap
  * 0202 is /boot
  * 0203 is /root
  * 0204 is /root
  * 0205 is /sapcds
  * 0206 is more of /sapcds
  * 0207 is more of /sapcds
  * 0208 is more of /sapcds
  * 0209 is /install
  * 0210 is /usr/sap/trans
  * 0211 is /usr/sap/M5D
  * 0212 is /sapmnt
  MDISK 0191 3390 981 20 515U1R MR
  MDISK 0201 3390 1 3338 DK8120 MR
  MDISK 0202 3390 1 200 DK8121 MR
  MDISK 0203 3390 1 3338 DK8122 MR
  MDISK 0204 3390 1 3338 DK8123 MR
  MDISK 0205 3390 1 3338 DK8124 MR
  MDISK 0206 3390 1 3338 DK8125 MR
  MDISK 0207 3390 1 3338 DK8126 MR
  MDISK 0208 3390 1 3338 DK8127 MR
  MDISK 0209 3390 1 3338 DK8128 MR
  MDISK 0210 3390 1 3338 DK8220 MR
  MDISK 0211 3390 1 3338 DK8221 MR
  MDISK 0212 3390 1 3338 DK8222 MR
```

We had to update our file system's export control file to allow root on other systems to be able to perform any commands necessary during installation. Example 30 on page 47 shows a sample of our /etc/exportfs file.
To ensure that we had valid TCP/IP definitions for each of our servers, we manually updated our /etc/hosts file on this system and updated the same file on each Linux on zSeries system. Example 31 shows our definitions for this system. This host file was used throughout the entire environment and replicated in every server.

Example 31 SAPCI /etc/hosts file

<table>
<thead>
<tr>
<th>Address</th>
<th>Host Name</th>
<th>Alias Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>127.0.0.1</td>
<td>localhost</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td># special IPv6 addresses</td>
</tr>
<tr>
<td>::1</td>
<td>localhost</td>
<td>ipv6-localhost ipv6-loopback</td>
</tr>
<tr>
<td>fe00::0</td>
<td></td>
<td>ipv6-localnet</td>
</tr>
<tr>
<td>ff00::0</td>
<td></td>
<td>ipv6-mcastprefix</td>
</tr>
<tr>
<td>ff02::1</td>
<td></td>
<td>ipv6-allnodes</td>
</tr>
<tr>
<td>ff02::2</td>
<td></td>
<td>ipv6-allrouters</td>
</tr>
<tr>
<td>ff02::3</td>
<td></td>
<td>ipv6-allhosts</td>
</tr>
<tr>
<td>9.12.10.32</td>
<td>T40Laptop.itso.ibm.com</td>
<td>t40laptop T40Laptop</td>
</tr>
<tr>
<td>9.12.4.97</td>
<td>sapci.itso.ibm.com</td>
<td>sapci</td>
</tr>
<tr>
<td>9.12.4.98</td>
<td>sapcm.itso.ibm.com</td>
<td>sapcm</td>
</tr>
<tr>
<td>9.12.4.99</td>
<td>sapdi.itso.ibm.com</td>
<td>sapdi</td>
</tr>
<tr>
<td>9.12.4.21</td>
<td>wtsc04oe.itso.ibm.com</td>
<td>wtsc04oe WTSC04OE</td>
</tr>
<tr>
<td>9.12.4.20</td>
<td>wtsc04.itso.ibm.com</td>
<td>wtsc04 WTSC04</td>
</tr>
<tr>
<td>9.12.4.100</td>
<td>idadm001.itso.ibm.com</td>
<td>idadm001 IDADM001</td>
</tr>
<tr>
<td>9.12.4.101</td>
<td>idadm002.itso.ibm.com</td>
<td>idadm002 IDADM002</td>
</tr>
</tbody>
</table>

The base SAP Dialog Instance Master (SAPDI)

The SAPDI system was installed via standard SAP installation tools. Refer to SAP R/3 Enterprise on UNIX: IBM DB2 Universal Database for z/OS and OS/390, Version 1, November 2003.

The SAPDI Master instance is a work area where the base configuration is installed and configured prior to copying to the Configuration Master Instance.
Your configuration may vary from this setup and may require additional resources.

We have configured and started the SNMP daemon on this system. This in effect ensures that the SNMP daemon is installed and autostarted on each of the dynamically provisioned clones without any additional work.

We configured a static virtual network address for our network adapter on this system as evidenced by the NICDEF statement in Example 32.

Example 32  SAPDI directory entry

```
USER SAPDI XXXXXXXX 512M 1024M G
PROFILE IBMDFLT
  SPOOL 000C 2540 READER *
  SPOOL 000D 2540 PUNCH A
  SPOOL 000E 1403 A
  CONSOLE 009 3215 T
  LINK MAINT 0190 0190 RR
  LINK MAINT 019D 019D RR
  LINK MAINT 019E 019E RR
  LINK MAINT 0402 0402 RR
  LINK MAINT 0401 0401 RR
  LINK MAINT 0405 0405 RR
  IPL CMS
  MACHINE ESA
  NICDEF C200 TYPE QDIO LAN SYSTEM LAN1 MACID 009999
  * 0201 IS SWAP 3
  * 0202 IS /BOOT 3
  * 0203 IS / (ROOT) 3
  * 0204 IS / (BOOT)
    MDISK 0191 3390 1021 20 515U1R MR
    MDISK 0C01 3390 1 3338 DK8223 MR
    MDISK 0C02 3390 1 400 DK8224 MR
    MDISK 0C03 3390 1 3338 DK8225 MR
    MDISK 0C04 3390 1 3338 DK8226 MR
```

Disk drive layout for SAPDI

The SAP Dialog Instance was installed as a normal SAP installation. The VM user direct entry depicted in Example 32 shows that we defined one VM minidisk for the guest 191 A disk, and four disk drives to hold the Linux file system and the SAP file system. Access to additional data was accomplished by mounting required disk drives from the SAPCI instance via NFS.

The first Linux on zSeries disk is 0C01, and it is assigned as the swap partition. Disk 0C02 is assigned to /boot. Disks 0C03 and 0C04 are assigned to /root. This makes up the base Linux file systems needed to start Linux.
To minimize the amount of disk utilized for this project, the SAP-specific file systems for this dialog instance are also contained within these four disk drives that hold the standard Linux operating system.

The IBM Dynamic Infrastructure for mySAP documentation states that SNMP must be enabled on all SAP systems. We configured the SNMP control file (snmpd.conf) to allow read-only access to Public and ensured that the SNMP daemon was automatically started at system boot. Example 33 shows actual coding of the text.

Example 33  SAPDI snmpd.conf entry

| rocommunity public default |

The IBM DI SAP Configuration Master Instance (SAPCM)

The SAPCM system was build by running VM scripts to clone the disk drives from SAPDI. These scripts can be referenced in *IBM Dynamic Infrastructure for MySAP Business Suite on IBM zSeries*, SG24-6473.

The SAP Configuration Master Instance is the clone source for building the dynamically provisioned servers. It is meant to be a static system that matches the SAP Dialog Instance Master System at all times. See “SAP base system overview” on page 9 for how we set up this system. Your configuration may vary from this setup and may require additional resources.

Since the Dialog Instance Master Server was configured for SNMP, this system will also be configured as such.

We configured a static virtual network address for our network adapter on this system as evidenced by the NICDEF statement in Example 34 on page 50.

**Disk drive layout for SAPCM**

The SAP Configuration Master Dialog Instance was created by cloning the SAPDI system and customized during boot by running the /etc/init.d/boot.local script at system load time.

The VM user direct entry depicted in Example 34 on page 50 shows that we defined one VM minidisk for the guest 191 A disk and four disk drives to hold the Linux file system and the SAP file system. Additional data was accessed by mounting required disk drives from the SAPCI instance via NFS.

The first Linux on zSeries disk is 0C01, and it is assigned as the swap partition. Disk 0C02 is assigned to /boot. Disks 0C03 and 0C04 are assigned to /root. This makes up the base Linux file systems needed to start Linux.
We also defined four minidisks to be used as the source drives for the IBM Dynamic Infrastructure for mySAP cloning process. They have the same naming convention as above except they begin with the number 1 instead of 0 (1C01, 1C02, 1C03, 1C04). The drives are also coded for access MR.

To minimize the amount of disk utilized for this project, the SAP-specific file systems for this dialog instance are also contained within these four disk drives that hold the standard Linux operating system.

Example 34  SAPCM directory entry

```
USER SAPCM XXXXXXXX 512M 1024M G
    PROFILE LINDFLT
    CLASS G
    STORAGE 512M
    MAXSTORAGE 2047M
    IPL CO2
    IUCV ALLOW
    MACHINE ESA
    CONSOLE 0009 3215 T
    SPOOL 000C 2540 READER *
    SPOOL 000D 2540 PUNCH A
    SPOOL 000E 1403 A
    LINK MAINT 0190 0190 RR
    LINK MAINT 0190 019D RR
    LINK MAINT 019E 019E RR
    IPL CMS
    MACHINE ESA
    NICDEF C200 TYPE QDIO LAN SYSTEM LAN1 MACID 010000
    * 0C01 IS SWAP
    * 0C02 IS /BOOT
    * 0C03 IS / (ROOT)
    * 0C04 IS / (ROOT)
    MDISK 0191 3390 1001 20 515U1R MR
    MDISK 1C01 3390 1 3338 DK8227 MR
    MDISK 1C02 3390 1 400 DK8228 MR
    MDISK 1C03 3390 1 3338 DK8236 MR
    MDISK 1C04 3390 1 3338 DK8237 MR
```

We wrote and installed the bash script to enable automatic modification of system HOSTNAME and TCP/IP address based on the name of the VM Guest. Note that this will occur only once, at first load time, due to the logic in the script. Additional customization is possible in this script and other boot scripts, as well as with the workflows provided by the IBM Dynamic Infrastructure for mySAP automation package. If we had had more time, we would have also looked into the ability to use the scripts mentioned in *IBM Dynamic Infrastructure Enterprise*
Example 35  SAPCM /etc/init.d/boot.local script

```bash
#!/bin/sh
VAR1=`cat /proc/sysinfo | grep 'VM00 Name' | awk '{print $3}'`
echo $VAR1
  if [ -e /install/chgfile.$VAR1 ]
    then
      echo fc exists $?
    else
      touch /install/chgfile.$VAR1
      file1=chgfile.$VAR1
      echo file1 is $file1
      cp /etc/HOSTNAME.$VAR1 /etc/HOSTNAME
      cp /etc/sysconfig/network/ifcfg-eth0.$VAR1
      /etc/sysconfig/network/ifcfg-eth0
      echo files copied rc $?
      fi
  echo end of file
```

Example 36 shows the fstab file defined for the SAPCM Dialog Instance.

Example 36  SAPCM fstab file

```
/dev/sysvg/syslv     /       reiserfs defaults 1 1
/dev/dasdb1        /boot    reiserfs defaults 1 2
/dev/dasda1        swap     swap      pri=42 0 0
devpts             /dev/pts  devpts    mode=0620,gid=5 0 0
proc                /proc proc            defaults 0 0
sapci:/sapmnt/M5D/profile /sapmnt/M5D/profile  nfs defaults 0 0
sapci:/sapmnt/M5D/global /sapmnt/M5D/global  nfs defaults 0 0
sapci:/usr/sap/trans  /usr/sap/trans  nfs defaults 0 0
```

The IBM Dynamic Infrastructure for mySAP documentation states that SNMP must be enabled on all SAP systems. We configured the SNMP control file (snmpd.conf) to allow read-only access to Public and ensured that the SNMP daemon was automatically started at system boot. Example 37 shows actual coding of the text.

Example 37  SAPCM snmpd.conf entry

```
rocommunity public default
```
Scenario execution

In this section we show an execution of an IBM Dynamic Infrastructure for mySAP Subscription operation. This is done according to our scenario description shown in “Integration scenario overview” on page 4.

The Subscription operation demonstrates that all of the configuration steps performed in our environment and described throughout this paper lead to a successful implementation of IBM Dynamic Infrastructure for mySAP and consolidates the integration with Tivoli Provisioning Manager.

To perform a Subscription operation, follow these steps:

1. Log on to the IBM Dynamic Infrastructure for mySAP GUI using defined user IDs. In our environment we defined user IDs in “Configuring LDAP security” on page 14. In this example we log on as dexterap.

2. On the main panel, select **Subscriptions**.
3. The window in Figure 5 opens. From the Select action pull-down menu, select **Create Subscriptions**.

![Figure 5  Subscriptions window](image)
4. The Create Subscriptions window is displayed (Figure 6). Select Create New Customer. Enter the customer name and click Create. In our environment we created a customer named ScenarioCustomer.

Fill out the subscription details. Our scenario used the following details:

<table>
<thead>
<tr>
<th>Rating package</th>
<th>Basic rating package</th>
</tr>
</thead>
<tbody>
<tr>
<td>Provisioned AppServer and DB server</td>
<td>AppServer: zVM/ zLinux;</td>
</tr>
<tr>
<td></td>
<td>Database: DB/ zSeries</td>
</tr>
<tr>
<td>Configuration</td>
<td>SubscriptionConfiguration</td>
</tr>
<tr>
<td>Customer SLA</td>
<td>End user response time less than 4 seconds for 98% of requests</td>
</tr>
</tbody>
</table>

Click Create.

![Create subscription window](image-url)
5. This starts the subscription action in IBM Dynamic Infrastructure for mySAP.

![Figure 7 Create subscription in progress](image1)

6. The subscription triggers the euisap.CreateServiceEnvironment workflow in Tivoli Provisioning Manager. DCM entries for Customer, Application, and Cluster are dynamically created in Tivoli Provisioning Manager by this workflow.

![Figure 8 Customer, Application, and Cluster definitions](image2)
7. A server provisioning operation is set off by the euisap.ClusterAddServer workflow, which selects a server from the resource pool and allocates to the Cluster. The selected server status changes to *in transition*.

![Figure 9 Allocated server in transition state](image)

8. The provisioning operation proceeds by performing a software stack installation on the selected server. This is executed by the zvm.bootServerInstallImage workflow.

9. The zvm.bootServerInstallImage workflow issues a series of workflows performing calls to the managed-through server of our scenario using the VMAPI. These calls are used to:
   - Connect and authenticate to our zVM system.
   - Perform queries on the available VM user IDs to host our predefined z/Linux software stack.
   - Create an entry on the zVM directory for the new user ID.
   - Allocate a series of minidisks to host the software stack image.
   - Install the software stack image.
   - Activate a new user ID. This action activates the zLinux image.

![Figure 10 VMAPI workflows](image)
10. The provisioning operation proceeds by verifying the software stack installation process. This is performed by the zvm.softwareCheckInstallationStatus workflow.

11. The zvm.sshExecuteCommandRsa workflow is executed. It monitors the activation of the newly provisioned server by executing the SNMPPing Java plug-in. When it receives a response, it marks the server as active.

<table>
<thead>
<tr>
<th>RequestId</th>
<th>Workflow Name</th>
<th>Start Time</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>12772</td>
<td>zvm.sshExecuteCommandRsa</td>
<td>September 16, 2005 6:00:17 PM EDT</td>
<td>in-progress</td>
</tr>
<tr>
<td>12771</td>
<td>Default_Device_Execute_Command</td>
<td>September 16, 2005 6:00:15 PM EDT</td>
<td>in-progress</td>
</tr>
<tr>
<td>12770</td>
<td>zvm.softwareCheckInstallationStatus</td>
<td>September 16, 2005 6:00:12 PM EDT</td>
<td>in-progress</td>
</tr>
<tr>
<td>12769</td>
<td>Group_Status_Updater</td>
<td>September 16, 2005 6:00:00 PM EDT</td>
<td>SUCCESS</td>
</tr>
<tr>
<td>12768</td>
<td>zvm.executeYapiCommand</td>
<td>September 16, 2005 5:59:22 PM EDT</td>
<td>SUCCESS</td>
</tr>
<tr>
<td>12767</td>
<td>Default_Device_Execute_Command</td>
<td>September 16, 2005 5:58:20 PM EDT</td>
<td>SUCCESS</td>
</tr>
<tr>
<td>12766</td>
<td>zvm.executeYapiCommand</td>
<td>September 16, 2005 5:58:12 PM EDT</td>
<td>SUCCESS</td>
</tr>
<tr>
<td>12765</td>
<td>Default_Device_Execute_Command</td>
<td>September 16, 2005 5:58:10 PM EDT</td>
<td>SUCCESS</td>
</tr>
<tr>
<td>12764</td>
<td>zvm.executeYapiCommand</td>
<td>September 16, 2005 5:58:03 PM EDT</td>
<td>SUCCESS</td>
</tr>
<tr>
<td>12763</td>
<td>Default_Device_Execute_Command</td>
<td>September 16, 2005 5:58:02 PM EDT</td>
<td>SUCCESS</td>
</tr>
</tbody>
</table>

Figure 11  Software stack installation verification workflows

12. The original euisap.CreateServiceEnvironment workflow starts a WaitforSubsystemState Java plug-in, which verifies the status of the SAP application on the provisioned server. When the SAP application is successfully started, the workflow proceeds by executing the StartMonitoring Java plug-in. This Java plug-in notifies IBM Dynamic Infrastructure for mySAP to start collecting measurement data for the subscription.
13. When the execution of the euisap.CreateServiceEnvironment is successfully completed, all of the provisioning operations in Tivoli Provisioning Manager are finalized.

Figure 12 shows a sequence of screen shots of the workflow execution history of our Tivoli Provisioning Manager. It shows a successful execution of the euisap.CreateServiceEnvironment workflow and all other workflows that play a role on the provisioning operations of the Subscription operation.

```
<table>
<thead>
<tr>
<th>RequestId</th>
<th>Workflow Name</th>
<th>Start Time</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>12772</td>
<td>zvm.sshExecuteCommandRsa</td>
<td>September 16, 2005 6:00:17 PM EDT</td>
<td>success</td>
</tr>
<tr>
<td>12771</td>
<td>Default_Device_Execute_Command</td>
<td>September 16, 2005 6:00:15 PM EDT</td>
<td>success</td>
</tr>
<tr>
<td>12770</td>
<td>zvm.softwareCheckInstallationStatus</td>
<td>September 16, 2005 6:00:12 PM EDT</td>
<td>success</td>
</tr>
<tr>
<td>12769</td>
<td>Group_Status_Updater</td>
<td>September 16, 2005 6:00:00 PM EDT</td>
<td>success</td>
</tr>
<tr>
<td>12768</td>
<td>zvm.executeVmapiCommand</td>
<td>September 16, 2005 5:59:22 PM EDT</td>
<td>success</td>
</tr>
<tr>
<td>12767</td>
<td>Default_Device_Execute_Command</td>
<td>September 16, 2005 5:58:20 PM EDT</td>
<td>success</td>
</tr>
<tr>
<td>12766</td>
<td>zvm.executeVmapiCommand</td>
<td>September 16, 2005 5:58:12 PM EDT</td>
<td>success</td>
</tr>
<tr>
<td>12765</td>
<td>Default_Device_Execute_Command</td>
<td>September 16, 2005 5:58:10 PM EDT</td>
<td>success</td>
</tr>
<tr>
<td>12764</td>
<td>zvm.executeVmapiCommand</td>
<td>September 16, 2005 5:58:03 PM EDT</td>
<td>success</td>
</tr>
<tr>
<td>12763</td>
<td>Default_Device_Execute_Command</td>
<td>September 16, 2005 5:58:02 PM EDT</td>
<td>success</td>
</tr>
<tr>
<td>12762</td>
<td>zvm.bootServerInstallImage</td>
<td>September 16, 2005 5:57:53 PM EDT</td>
<td>success</td>
</tr>
<tr>
<td>12761</td>
<td>zvm.imageSoftwareStackInstallation</td>
<td>September 16, 2005 5:57:51 PM EDT</td>
<td>success</td>
</tr>
<tr>
<td>12760</td>
<td>euisap.ClusterAddServer</td>
<td>September 16, 2005 5:57:35 PM EDT</td>
<td>success</td>
</tr>
<tr>
<td>12759</td>
<td>Default_Device_Execute_Command</td>
<td>September 16, 2005 5:57:10 PM EDT</td>
<td>success</td>
</tr>
<tr>
<td>12758</td>
<td>zvm.executeVmapiCommand</td>
<td>September 16, 2005 5:56:03 PM EDT</td>
<td>success</td>
</tr>
<tr>
<td>12757</td>
<td>euisap.ClusterCleanupServer</td>
<td>September 16, 2005 5:52:38 PM EDT</td>
<td>success</td>
</tr>
<tr>
<td>12756</td>
<td>euisap.CreateServiceEnvironment</td>
<td>September 16, 2005 5:21:03 PM EDT</td>
<td>success</td>
</tr>
</tbody>
</table>
```

*Figure 12  euisap.CreateServiceEnvironment workflow - successful execution*
14. The newly provisioned server is finally ready for attending requests to the SAP application it hosts, and its status in Tivoli Provisioning Manager is changed to *available*.

![Figure 13 Allocated server in available state](image13)

15. IBM Dynamic Infrastructure for mySAP changes the subscription state to *allocated*.

![Figure 14 Subscription status panel](image14)
The team that wrote this Redpaper

This Redpaper was produced by a team of specialists from around the world working at the International Technical Support Organization, Austin Center.

**Edson Manoel** is a Certified IT Specialist at IBM working in the ITSO, Austin Center, in the systems management area. Prior to joining the ITSO, Edson worked in the IBM Software Group, Tivoli Systems, and in IBM Brazil Global Services Organization. He was involved in numerous projects in designing and implementing systems management solutions for IBM clients and Business Partners. Edson holds a Bachelor of Science degree in applied mathematics from Universidade de Sao Paulo, Brazil.

**Patrick Horkan** is an IBM Field Technical Sales Specialist for SAP on zSeries DB2. He has worked with both SAP R/3 and SAP BW over the past 10 years and is one of the technical sales leads for the support of IBM Dynamic Infrastructure for the Americas. His areas of expertise include assisting customers planning for and implementing complex multi-SAP application solutions when high availability or continuous operations are required. Patrick has 21 years of information technology experience.

**Lydia Parziale** is a Project Leader for the ITSO team in Poughkeepsie, New York, with domestic and international experience in technology management including software development, project leadership, and strategic planning. Her areas of expertise include e-business development and database management technologies. Lydia has been employed by IBM for 22 years in various technology areas.

Thanks to the following people for their contributions to this project:

Budi Darmawan  
International Technical Support Organization, Austin Center

Peter Taube  
IBM Systems &Technology Group, IBM Dynamic Infrastructure L3 Service

Gerold Kurth  
IBM Systems &Technology Group, IBM Dynamic Infrastructure Development

Albert D. Rodi  
IBM Sales and Distribution, Certified Technical Sales Specialist for SAP
Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:
IBM Director of Licensing, IBM Corporation, North Castle Drive Armonk, NY 10504-1785 U.S.A.

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE: 
This information contains sample application programs in source language, which illustrates programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. You may copy, modify, and distribute these sample programs in any form without payment to IBM for the purposes of developing, using, marketing, or distributing application programs conforming to IBM's application programming interfaces.
Send us your comments in one of the following ways:

- Use the online Contact us review redbook form found at: ibm.com/redbooks
- Send your comments in an e-mail to: redbook@us.ibm.com
- Mail your comments to:
  IBM Corporation, International Technical Support Organization
  Dept. JN9B  Building 905, 11501 Burnet Road
  Austin, Texas 78758-3493 U.S.A.

Trademarks

The following terms are trademarks of the International Business Machines Corporation in the United States, other countries, or both:

- DirMaint™
- Distributed Relational Database Architecture™
- DB2 Connect™
- DB2®
- DRDA®
- IBM®
- Redbooks™
- Redbooks (logo)™
- Tivoli®
- WebSphere®
- xSeries®
- z/OS®
- z/VM®
- zSeries®

The following terms are trademarks of other companies:

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Microsoft, Windows, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

Other company, product, or service names may be trademarks or service marks of others.