Maintaining Prime Service Catalog

This chapter contains the following topics:

Maintaining Prime Service Catalog

This chapter describes about startup and shutdown procedures for application components, recommended backup practices, configuration management and customizations of application components, ongoing maintenance tasks, and critical error conditions, error messages and resolutions

Note The designation <APP_HOME> indicates the root directory where Service Catalog is installed.

System Hardening

Linux Puppet Master, Linux, and Windows target VMs must have direct Internet access or through a proxy. You can alternatively configure the VM to connect with an internal repository.

The Gateway VM must be configured to route traffic as mentioned in (any specific Guide) and should not be restricted after configuration.

You must not implement firewall or SELinux rules to block specific ports after installing Puppet Master or Agents. For more information about the ports specific to applications, see (needs citation).

Note It is not recommended that you close ports in the target VM for application install. You can open ports in the Gateway VM as mentioned in Cisco Prime Collaboration Quick Start Guide.

On Linux VMs hosting Puppet Master installed with SELinux, the default installation and configuration opens the required ports (8140, 8139, and 5150) long with IPTables rules.

The Puppet Agent does not alter port access or security protocols in the VMs. Hardening these devices can cause the related applications to fail during install or run. VMs are placed in fenced containers in the UCS Director and access is controlled via a Gateway VM to prevent application failures.

Performing Backup

The components of a fully deployed system include Service Catalog, Integration Server (Service Link), and Advanced Reporting (Cognos). Service Catalog and Integration Server are deployed to the application server in the Service Catalog.war and ISEE.war deployment packages, respectively.

Note We recommend backing up each component as it is deployed, and saving any customizations as they are developed or modified.

- Backup operation must be scheduled regularly.

The following databases must be backed up:

- Transactional database (by default, Service Catalog)- Contains not only production data but also metadata for configuring services, service components, and other application objects.
- Analytical database- Contains data for building the standard reports, as well as the Service Catalog and Demand Center data marts.
- Content Store database- Contains user-generated content available in the business view of the reporting environment. Such content includes the definitions of all reports, both those provided by Service Catalog and those written by Advanced Reporting users, as well as report views, schedules, and saved reports generated from any reports.

Tuning the Application Server

The following tuning suggestions are applicable to many Servce Catalog sites. For additional tuning suggestions, see the documentation specific to your application server.

Configuring Service Catalog Compression

If your organization has a significant number of distant users, it will make sense to turn on GZIP compression (RFC 1952) for HTTP responses, see RFC 2616:

- Section 3.5: Content-coding
- Section 14.3: Accept-Encoding
- Section 14.11: Content-Encoding

GZIP compression will benefit users working over slow or high latency networks. However, GZIP compression will add a slight overhead on both the server and the user's browser.

To enable GZIP compression:

Step 1 Locate the web.xml under RequestCenter.war/WEB-INF. For example, a typical location is:

Example:

- `C:\jboss-as-7.1.1.Final\ServiceCatalogServer\deployments\RequestCenter.war\WEB-INF`

Step 2 Look for the following entry (which is commented out):

Example:

```
<!--filter><filter-name>CompressingFilter</filter-name><filter-class>com.planetj.servlet.filter.compression.CompressingFilter
</filter-class></filter-->
```

Step 3 Remove the comments, so the entry becomes:

Example:

```
<filter><filter-name>CompressingFilter</filter-name><filter-class>com.planetj.servlet.filter.compression.CompressingFilter
</filter-class></filter>
```

Step 4 Look for the following entry (which is commented out):

Example:

```
<!--filter-mapping
id="newscale_gzip_filter_1"><filter-name>CompressingFilter</filter-name><url-pattern>/*</url-pattern>
</filter-mapping-->
```

Step 5 Remove the comments, so the entry becomes:

Example:

```
<!--filter-mapping
id="newscale_gzip_filter_1"><filter-name>CompressingFilter</filter-name><url-pattern>/*</url-pattern>
</filter-mapping-->
```

Step 6 Save the file and restart the application servers.

Changing the Java Memory Settings

Java memory settings are specific to the Java Virtual Machine (JVM) used by the application server. Use the commands "java –h" and "java –X" to return a full listing of the options available on your system. Ensure that you are calling the same JVM that is used by your application server when issuing these commands.

- -ms –mx as appropriate (usually 1GB of memory is reserved for the heap within the JVM).
- -server mode is recommended for Oracle JVM.
- A common modification is to increase the garbage collector's maximum permanent generation size to 128MB with the argument: --XX:MaxPermSize=128m.

The Java memory switches governing the minimum and maximum heap size available to the JVM may need to be tuned if Service Catalog encounters “out of memory” errors. For example, on Weblogic the following settings have been successfully applied.

```
MEM_ARGS="-verbose:gc -Xms1024m -Xmx1024m-XX:+PrintGCTimeStamps -XX:+PrintGCDetails -XX:MaxPermSize=256m"
```

JMS Queue Connection Factory Settings

The number of connections for the queue connection factory should be configured based on the work load on the JMS server. The recommended setting for a single Service Catalog instance is 25. There is no hard and fast rule on the number of connections required based on the number of servers in the cluster. Some tuning effort may be required to arrive at the optimal connection settings for the application environment.

Upgrading/Replacing the JDK

You can upgrade the JDK to a later version by following the steps below:

- Edit the script named “setEnv.cmd” on the <APP_HOME>/bin directory to specify the path to the new JDK.
- For customers using the startup scripts, save the revised setEnv.cmd file and then restart the server.
- For customers using the Windows services, stop the windows services, uninstall the window services (using the <APP_HOME>\bin\uninstall*.cmd scripts), and then re-install the window services again (using the <APP_HOME>\bin\install*.cmd scripts).

Tuning the Database

We can list a few of the most frequently asked questions regarding how to configure and tune Service Catalog databases and the answers to those questions. For more details on these issues, you will need to see the appropriate database-specific documentation. Many of these FAQs pertain to Oracle which has more opportunities for tuning than does SQLServer.

- For both Oracle and SQLServer, experts recommend installing the database files on a RAID 1+0 (striped + mirrored) disk, rather than on RAID 5, which is the preferred choice for software installation.
- An Oracle database should be configured to use locally managed tablespaces (LMT) and Automatic Segment Space Management (ASSM). These technologies eliminate previous difficulties with improperly specified table or tablespace parameters (PCTUSED, PCTFREE, INITIALEXTENT, NEXTEXTENT).
- Use different databases/instances for the OLTP Service Catalog and OLAP database (standard reports and the Service Catalog data marts). In Oracle releases prior to 10g, this was required in order to create tablespaces with different block sizes. Even in 10g and beyond, it is recommended so that configuration parameters can be adjusted to the vastly different activities in OLTP vs. OLAP databases. Oracle DBAs are urged to read Oracle's excellent documentation on Database Administration for Data Warehouses.

Specific Recommendations for Service Catalog

- For the OLTP database, create a primary tablespace named REQUESTCENTER. Allow for 10 MB per user, with a minimum size of 500 MB, for the tablespace. Your database administrator should choose an extent management strategy that fits well with the best practices of your organization.
- A very rough estimate of database storage required is 500 KB for each requisition completed. This varies greatly with the complexity of the service form, the authorization structure, and the delivery plan.
- Sites with many Service Link tasks will notice significant growth in the database size, attributable to storing Service Link messages. Recent versions of Service Catalog have included increasingly effective compression algorithms for these messages, as well as a means to configure message context. Additional details are available in the Cisco Prime Service Catalog Integration Guide. Database scripts for purging Service Link messages for completed tasks are available as stored procedures in the RequestCenter database and can be executed either as a one-time job or on a recurring basis.

Tuning Oracle

You can optimize performance for Oracle database in the following ways:

- Gather statistics on the OLTP database (both tables and indexes) on a regular basis. This can be automated via Oracle Enterprise Manager (OEM).
- Perform column-level histogram analysis to further optimize the Service Manager indexes.
- Gather statistics on the Service Catalog data marts after the data marts have been refreshed.
- Review table allocation, tablespace fragmentation, and row chaining.
- Grant access to the SELECT_CATALOG_ROLE for monitoring query performance.

Apply settings similar to the following:

Table 1: Oracle settings

Parameter	Value
perf.__large_pool_size	16777216
*.processes	300
*.pga_aggregate_target	1059145600
*.sga_max_size	716582400 #internally adjusted
*.sga_target	716582400
*.sort_area_size	500000000

Gather Statistics on the Database

Use the DBMS_STATS.GATHER_SCHEMA_STATS command to gather statistics on all tables and indexes in the RequestCenter database. In the example below, “RC User” is the schema owner.

```
execute DBMS_STATS.GATHER_SCHEMA_STATS(ownname=>'RCUser',cascade=>TRUE);
```

Histogram Analysis

The Oracle Database Administration chapter on “Managing Optimizer Statistics” recommends:

- When gathering statistics on a table, DBMS_STATS gathers information about the data distribution of the columns within the table. The most basic information about the data distribution is the maximum value and minimum value of the column. However, this level of statistics may be insufficient for the optimizer's needs if the data within the column is skewed. For skewed data distributions, histograms can also be created as part of the column statistics to describe the data distribution of a given column.
- Histograms are specified using the METHOD_OPT argument of the DBMS_STATS gathering procedures. Oracle Corporation recommends setting the METHOD_OPT to FOR ALL COLUMNS SIZE AUTO. With this setting, Oracle automatically determines which columns require histograms and the number of buckets (size) of each histogram. You can also manually specify which columns should have histograms and the size of each histogram.

The tables for which it is critical to gather histogram-level statistics are:

- TxActivity
- TxProcess
- TxRequisition
- TxRequisitionEntry
- DirPerson
- DirOrganizationalUnit
- UIEntry

A sample DBMS_STATS command for collecting the statistics on each table would like look:

```
BEGIN
  DBMS_STATS.GATHER_TABLE_STATS (OWNNAME => 'RCUser',       TABNAME => 'TXACTIVITY',
      METHOD_OPT => 'FOR ALL COLUMNS SIZE AUTO');
END;
```

Tuning SQLServer

Enable snapshots with this command:

```
ALTER DATABASE <database name> SET READ_COMMITTED_SNAPSHOT ON
```

We recommend a SQLServer DBCC Reindex command, especially on volatile Service Catalog tables. The process should be regularly scheduled, typically weekly, at off hours.

The following tables are the most volatile and should be subject to DBCC Reindex.

Table 2: SQLServer commands

TxActivity	TxEventTriggerParam	TxPerformerSummary
TxActivityAssignment	TxIncident	TxProcess
TxAttribute	TxInternalOptionList	TxRequisition
TxCheckList	TxInvocation	TxRequisitionEntry
TxChecklistEntry	TxInvocationAttribute	TxRequisitionStep
TxComments	TxJMSMessage	TxRole
TxCondition	TxJoin	TxRule
TxDictionaryHTMLBindings	TxMultivalue	TxSatisfaction
TxDocument	TxObjectDataHTML	TxService
TxEmailSent	TxObjectDictionaries	TxSubscription
TxEventTrigger	TxObjectRelation	TxTimer

Sizing Cognos Database Components

Cognos maintains the definitions of all reports and queries in a database called the ContentStore. The Cognos KnowledgeBase includes entries on sizing and maintaining the ContentStore. Of particular interest are the formulas published for determining the size required for the ContentStore, based on estimated usage statistics.

A spreadsheet incorporating these formulas is available from the Cisco Technical Assistance Center (TAC). A sample is shown below.

Table 3: Cognos Database components

Component	# Estimated	Space per Unit (KB)	Total (KB)
Active Users	250		
Concurrent Users executing reports (Temporary disk space requirements)	50	100,000	5,000,000
Saved Reports 1-10 pages (2 per user, 1-Public, 1 – MyFolder)	500	340	170,000
Saved Reports 10-100 pages (9 per user, 4-Public, 5 – MyFolder)	2250	440	990,000
Saved Views 1-100 rows (3 per user – all MyFolders)	750	250	187,500

Component	# Estimated	Space per Unit (KB)	Total (KB)
Saved Views 100-1000 rows (8 per user – all MyFolders)	2000	350	700,000
Folders Public MyFolders (5 per user)	1,250		0
FrameMaker Models (provided by Cisco)			20,000
Empty Content Store	1	3,000	3,000
Active Schedules (50 Day + 125 Weekly)	175	30	5,250
Total			7,075,750

OLTP Database Tables

The transactional database consists of a set of relational tables that use a prefix naming convention. The following table is provided as an aid to DBAs or others who need to maintain or tune a production database. The structure and contents of these tables is proprietary to Cisco, which reserves the right to freely change table names or structures from release to release.

Table 4: OLTP Database table

Prefix	Meaning	Usage
Cnf	Configuration	Tables which contain internal configuration information used by Service Catalog; typically, these tables are small and their contents static in a production environment.
Co	Portal Content	Tables which contain Portal Manager Content and Page definitions.
Def	Definition	Tables which hold user definitions of user-configurable objects such as service forms, dictionaries, and checklists; table size varies with the size of the implementation, but is relatively stable in a production environment, typically changed only via usage of the Catalog Deployer.

Prefix	Meaning	Usage
Dir	Directory	Tables containing person and organizational information; table size for most is quite small (skills, projects, functional positions) and stable; those relating to persons vary greatly per organizational size.
JMS	Java Message Service	Internal Usage.
Mdr	Meta-data Repository	Tables containing meta-data for tables with a (user-defined) dynamic schema (for example. service items, standards, and portal).
Si	Service Item	Tables containing data for service items.
St	Standards	Tables containing data for standards.
Tx	Transaction	Tables which contain all transactions. Tables can be quite large and volatile.
Uc	User Content	Tables containing Portal Manager custom content.
UI	User Interface	Tables which define user-specific customizations for the user interface, such as Service Manager views, the default module that appears on login, and Service Link filters.
Xtr	External	Tables used by Service Link to manage external tasks; the definition tables (XtrDef) may be quite small, but the tables containing messages for external tasks are large and quick-growing.
XtrEUI	External End User Integration	Tables used for Directory Integration definitions.

Optimizing Performance through Purging and Partitioning

You can optimize performance through partition and purging.

Improving Performance through Historical Requisitions Partitioning

Historical Requisition Partitioning feature moves completed requisitions, namely, requisitions that have "Closed", "Canceled", "Delivery Canceled" or "Rejected" status, to historical transaction tables. The use of Historical Requisition Partitioning provides overall application performance improvement as a result of reducing the amount of data in the current transaction tables. The improvement can be seen in the filter and search of tasks, requisitions and external messages in the Service Manager, My Services and Service Link modules. ETL and request workflow processing will also benefit from the smaller population of data in the current transaction tables.

Historical Requisition Partitioning is controlled by the system setting "**Enable Historical Requisitions Scheduler**" in the Administration module. When it is enabled, requisitions that have been completed for more than 365 days are migrated by a background process to the historical transaction tables. The 365-day retention period is configurable and may be modified based on the specific needs of your organization.

You can execute the migration process of historical requisitions on an ad-hoc basis in the **Administration > Utilities** page when the scheduler is disabled.

Thus, requisition views in both My Services and Service Manager are separated into "**Recent**" and "**Historical**" views. Requisitions migrated to the historical transaction tables can be made accessible on when you select the **Enable Historical Requisitions View** system setting. These requisitions are displayed under **Historical** tab on the **My Services > Requisitions** page, as well as **Historical Requisitions** view in Service Manager. You cannot view tasks and external messages associated with the historical requisitions through UI, although the data is stored within the Service Catalog database.

Preparing for Historical Requisition Partitioning

The first-time execution of the historical requisition migration will likely cover a large amount of data. To reduce the impact on application users, execute the process manually from the Administration module during off-peak periods:

Navigate to Administration module and ensure that the **Enable Historical Requisitions Scheduler** setting is turned off. Under **Utilities**, go to the **Run Processes**, and enter the desired cut-off date. Optionally, specify the batch size and maximum number of requisitions to process.

A larger batch size shortens the processing time but requires higher amount of temporary space or rollback segment in the database server. Setting the maximum number of requisitions or clicking **Stop** allows you to limit the duration of historical requisition migration process. The processing rate and duration vary based on the average size of the requisitions.

Note Before executing the migration process, we recommend you to work with the database administrators and perform trial runs to estimate the time required for the first-time execution.

Considerations for Datamart

Historical transaction data are not extracted by the ETL process, for Reporting. Consider the following for configuring the historical requisition partitioning feature:

- ETL process is normally scheduled to run on a frequent basis. Hence requisition data is captured into Datamart prior to their migration to the historical tables. To ensure there is no data loss in the Datamart, set the historical requisition retention period to be greater than the frequency of the ETL process. For example, if ETL is set to run every 30 days, the historical requisition retention period should be set to 31 days or more.
- The process that migrates historical transaction data is automatically put on hold when it detects that the most recent ETL process timestamp is earlier than the cutoff date. For example, if the last ETL execution was on May 1st 12pm and the migration is going to select requisitions completed before May 1st 12:30pm, the migration process will exit immediately. This ensures that data are kept in the current transaction tables for extraction into Datamart before they get migrated.
- If your organization has the need to rebuild Datamart occasionally to capture backdated data (for example, making a dictionary or service reportable after the fact), the changes will not take effect on historical transactions by re-running the ETL process. In fact, the historical data will not be recoverable in the Datamart once they have been purged. To ensure such Datamart rebuild process is still possible, configure the data retention period to a duration that has provisions for backdated changes. In addition, the Datamart should no longer be emptied in a rebuild process for the reason above. Only the portion of data that are still available in the current transaction tables can be deleted and re-inserted into the Datamart during the re-execution of ETL.

Key Settings for Historical Requisition Processing

The following settings are applicable for historical requisition partitioning:

1. newscale.properties file (located in the RequestCenter.war/WEB-INF/classes/config directory)
2. reqArchival.poller.cron - This controls the frequency of the background process that migrates historical data. It uses the standard cron syntax and is scheduled to run every 30 minutes.
3. reqArchival.process.maxRecords - This controls the maximum number of requisitions to be processed in each run. A higher number may be set if the process is intended to be run for a longer period of time during scheduled maintenance.
4. reqArchival.cutOffDate.days - This controls the retention period of completed requisitions in the current transaction tables. By default, the retention period is set to 365 days.
5. reqArchival.process.batchSize - This controls the number of historical requisitions included during each database commit. A larger batch size will shorten the processing time but will require higher amount of temp space or rollback segment in the database server.

1. support.properties file (located in the RequestCenter.war/WEB-INF/classes/config directory)

reqArchival.poller.enable - This controls whether the application instance can be used to run the historical requisition migration process. In a clustered Service Catalog environment, only one of the nodes will be used to execute the migration process at any time. One or more of the nodes may have this property set to "false" if the server is a less powerful machine or is meant for disaster recovery purpose.

Other settings in the above files should remain unchanged under normal circumstances.

Purging Requisitions

Service Catalog provides a transaction purge feature to delete transactions older than a chosen date or those that meet other user-specified criteria. This allows the application administrator to remove test requisitions before deleting test users and sample services. However, through the purge feature, you cannot perform mass data deletion. Also, you must avoid prolonged maintenance phase.

Software Requirements

- Database client for executing purge scripts
- sqlplus, for Oracle, must be installed and configured to connect to the RequestCenter database;
- osql, for SQL Server.

Preparing the System for Purging

Step 1 Make a backup of the RequestCenter database before executing the purge scripts.

Step 2 Stop the Service Catalog and Service Link services when the purge scripts are executed.

Step 3 Locate the utility in:
<APP_HOME>**schema****util****purge**

If the machine where <APP_HOME> resides has the database client software, then you can execute the purge scripts from that machine. Otherwise, copy the entire **purge** folder to the machine where the RequestCenter database is located, or to another machine that has the prerequisite database client.

Step 4 Verify that the **purge** folder contains the following files:

- AddPurgeFilter.bat
- AddPurgeFilter.sh
- ClearAllPurgeFilter.bat
- ClearAllPurgeFilter.sh
- PurgeRequisitions.bat
- PurgeRequisitions.sh

Step 5 Execute the .bat files if you are on Windows Operating System, or the .sh files if you are on UNIX or Linux Operating System.

Caution If you have applied any Service Catalog service packs, repeat Step 3 (above), to ensure that you use the latest version of the purge scripts, as the scripts may be modified as part of the service packs.

Purging Requisition

Purging requisition consists of the following steps:

Step 1 Clear purge filter criteria.

Step 2 Configure purge filter criteria.

Step 3 Perform a dry run for the requisition purge.

Step 4 Perform the actual requisition purge.

Clearing Purge Filter Criteria

This step is not required if the same filter criteria are always used for purging requisitions (for example, purge all canceled requisitions). However, we recommend that the criteria from the previous run are cleared initially to avoid confusion.

Use the **ClearAllPurgeFilter** script to clear one or all filter criteria. If [*Purge Filter Name*] is not given, the script will remove all filter entries from the **CnfPurgeFilter** table in the RequestCenter database. Otherwise, the script removes only the specified [*Purge Filter Name*] if it exists in the **CnfPurgeFilter** table.

Oracle:

ClearAllPurgeFilter ORACLE [*SID*] [*User*] [*Password*] [*Purge Filter Name* (optional)]

SQL Server:

ClearAllPurgeFilter SQLSERVER [*Server*] [*Database*] [*User*] [*Password*] [*Purge Filter Name* (optional)]

Possible values for the optional [*Purge Filter Name*] are:

- CREATIONSTARTDATE
- CREATIONENDDATE
- CLOSEDSTARTDATE
- CLOSEDENDDATE
- REQUISITIONSTATUS
- REQUISITIONID
- REQUISITIONRANGE
- SERVICEID
- SERVICENAME

Adding Purge Filter Criteria

Use the **AddPurgeFilter** script to add one or more filter criteria. Requisitions will be deleted only if they meet all the purge criteria. The filter criteria are stored in the table **CnfPurgeFilter** in the RequestCenter database.

Use the following syntax appropriate for your database type:

- [*SID*] is the ORACLE_SID for Oracle database
- [*Server*] is the SQL Server database server name
- [*User*] is “RCUser”

- [*Password*] is the password for “RCUser”
- Refer to the parameters table for possible values for [*Purge Filter Name*] and [*Purge Filter Value*]

Oracle:

AddPurgeFilter ORACLE [*SID*] [*User*] [*Password*] [*Purge Filter Name*] [*Purge Filter Value*]

SQL Server:

AddPurgeFilter SQLSERVER [*Server*] [*Database*] [*User*] [*Password*] [*Purge Filter Name*] [*Purge Filter Value*]

Table 5: Purge filter criteria

Purge Filter Name	Description	Purge Filter Value
CREATIONSTARTDATE	Purge requisitions created on or after this date.	Date in DD-MON-YYYY format.
CREATIONENDDATE	Purge requisitions created on or before than this date.	Date in DD-MON-YYYY format.
CLOSEDSTARTDATE	Purge requisitions closed on or after this date.	Date in DD-MON-YYYY format.
CLOSEDENDDATE	Purge requisitions closed on or before than this date.	Date in DD-MON-YYYY format.
REQUISITIONSTATUS	Purge requisitions with the specified status.	Possible values are PREPARATION, OPEN, ONGOING, CLOSED, CANCELLED, REJECTED, DELIVERY CANCELLED, ORDERED or ALL.
REQUISITIONID	Purge a specific requisition based on the Requisition ID.	Unique number assigned to the requisition.
REQUISITIONRANGE	Purge specific requisitions based on the Requisition ID range.	The starting and ending Requisition ID with a dash in between; for example, 30001-39999.
SERVICEID	Purge requisitions that contain a specific service based on the Service ID.	Unique identifier of the service, as displayed on the Service Designer General page for the service definition.

Purge Filter Name	Description	Purge Filter Value
SERVICENAME	Purge requisitions that contain a specific service based on the Service Name. For SERVICEID and SERVICENAME filters, the complete requisition is deleted—including all service requests. Purge is at the requisition-level, not at the individual entry-(service) level.	Service Name enclosed in double quotes, for example, “Email Service”. **Note** This purge filter value must be an exact match, and is case-sensitive. **Note** On UNIX or Linux operating systems, do not use this purge filter if the Service Name contains spaces.

Validating Purge Filter Criteria

Before purging requisitions, optionally perform a “dry run” to check requisitions that would be removed, without actually deleting them. This will serve as a validation for filter criteria.

Use the **PurgeRequisitions** script to get a list of requisitions which meet the filter criteria.

Oracle:

PurgeRequisitions ORACLE [*SID*] [*User*] [*Password*] **DRY_RUN** [UserName]

SQL Server:

PurgeRequisitions SQLSERVER [*Server*] [*Database*] [*User*] [*Password*] **DRY_RUN** [UserName]

UserName is the Service Catalog login name of the person executing the script.

The list of requisitions found in the dry run is stored in the **LogPurge** table in the RequestCenter database. The log entries are appended to the table with a RunID incremented by one, for every execution. You can review the requisitions to be purged by querying the LogPurge table entries with the highest RunID.

The **LogPurge** table can grow quickly over time, if you perform many dry runs and requisition purges. Therefore, we recommend that you manually truncate the **LogPurge** table periodically to remove entries from previous runs.

You can repeat Steps 1 to 3 to revise the purge criteria. After the purge filter criteria have been finalized, you can proceed with the actual requisition purge.

Purging Requisitions based on Filter Criteria

The requisition purge removes those requisitions that meet the purge filter criteria and all transactional data associated with those requisitions, including tasks and Service Link messages.

Results from the actual requisition purge are also appended to the **LogPurge** table in the RequestCenter database. To perform the actual requisition purge, use the command **PurgeRequisitions** with the PURGE parameter as shown below:

Oracle:

PurgeRequisitions ORACLE [*SID*] [*User*] [*Password*] **PURGE** [UserName]

SQL Server:

PurgeRequisitions SQLSERVER [*Server*] [*Database*] [*User*] [*Password*] **PURGE** [UserName]

UserName is the Service Catalog login name of the person executing the script.

Purging Temporary Data

The workflow purge utility removes temporary data from the database related to workflow processing. Those data are no longer used in the product and can be removed to reduce the database size. Executing the purge utility periodically could also provide overall performance improvement.

The workflow purge utility is provided in the form of a stored procedure in the RequestCenter database. The purge utility can require an hour or more to execute if you have a large database. Hence the purge should be done during system down time or a low activity time window. We recommend a practice run on sandbox environment to establish duration of the script execution for your database.

To track the start/end times for the purge, enable the setting for displaying print statements in the SQL tool before you execute the stored procedure.

Purging Workflows using the Utility on Oracle Database

To run the utility on Oracle database:

Step 1 Back up the RequestCenter database.

Step 2 Use a query tool appropriate for your database (for example, SQL*Plus) and connect to RequestCenter database as the RCUser.

Step 3 Execute the following commands:

- SET SERVEROUTPUT ON
- EXECUTE sp_PurgeWorkflowTables ([FromDate], [ToDate], [UserName]);

Dates must be in DD-MON-YYYY format. UserName is the Service Catalog login name of the person executing the script.

At the end of the execution, the total elapsed time should be displayed. Here is an example of the output:

Example:

```
Creation/Data population of TxReq_temp-    Successful
Time taken for TxReq_Temp   : .17 s
Creation/Data population of TxReqEntry_temp-    Successful
Time taken for TxReqEntry_Temp   : .08 s
Creation/Data population of TxSubscription_Temp -    Successful
Time Taken for TxSubscritpion   : 5.39 s
Creation/Data population of TxProcess_Temp -    Successful
Creation/Data population of TxJoin_Temp -    Successful
Time Taken for TxJoin   : .91 s
Creation/Data population of TxCondition_Temp -    Successful
Time Taken for TxCondition   : 1.18 s
Creation/Data population of TxActivity_Temp -    Successful
Creation/Data population of TxEventTrigger_Temp -    Successful
Creation/Data population of TxEventTriggerParam_Temp -    Successful
Time Taken for TxEventTriggerParam   : .33 s
***Creation/Data population of TxEventTrigger -    Successful***
***Creation/Data population of TxProcess -    Successful***
Creation/Data population of XtrChannelInfo_Temp -    Successful
Creation/Data population of XtrChannelParameterSpec_Temp -    Successful
***Creation/Data population of XtrChannelParameterSpec -    Successful***
Elapsed time: 10.62 s
PL/SQL procedure successfully completed.
```

Purging Workflows using the Utility on SQL Server Database

To run the utility on SQL Server database:

Step 1 Back up the RequestCenter database.

Step 2 Use a query tool appropriate for your database (for example SQL Server Management Studio) and connect to RequestCenter database as the RCUser.

Step 3 Execute the following command:

- EXECUTE sp_PurgeWorkflowTables [FromDate], [ToDate], [UserName]

Dates must be in DD-MON-YYYY format. UserName is the Service Catalog login name of the person executing the script.

At the end of the execution, the total elapsed time should be displayed. Here's an example of the output:

Example:

```
(2258 row(s) affected)
Creation/Data population of TxReq_Temp-    Successful
Time taken for TxReq_Temp   : 0 s
(2639 row(s) affected)
Creation/Data population of TxReq_Temp-    Successful
Time taken for TxReqEntry_Temp   : 0 s
(56580 row(s) affected)
(0 row(s) affected)
(56580 row(s) affected)
Creation/Data population of TxSubscription_Temp -    Successful
Time taken for TxSubscription_Temp   : 6 s
(4551 row(s) affected)
(2 row(s) affected)
Creation/Data population of TxProcess_Temp -    Successful
Time taken for TxProcess_Temp   : 0 s
(4154 row(s) affected)
(0 row(s) affected)
(4154 row(s) affected)
Creation/Data population of TxJoin_Temp -    Successful
Time taken for TxJoin_Temp   : 1 s
(9382 row(s) affected)
(9382 row(s) affected)
Creation/Data population of TxCondition_Temp -    Successful
Time taken for TxCondition_Temp   : 2 s
(7017 row(s) affected)
Creation/Data population of TxActivity_Temp -    Successful
Time taken for TxActivity_Temp   : 0 s
(5528 row(s) affected)
Creation/Data population of TxEventTrigger_Temp -    Successful
Time taken for TxEventTrigger_Temp   : 0 s
(1202 row(s) affected)
Creation/Data population of TxEventTriggerParam_Temp -    Successful
Time taken for TxEventTriggerParam_Temp   : 0 s
(5528 row(s) affected)
***Creation/Data population of TxEventTrigger -    Successful***
(1202 row(s) affected)
***Creation/Data population of TxEventTriggerParam -    Successful***
(4553 row(s) affected)
***Creation/Data population of TxProcess -    Successful***
(645 row(s) affected)
```

```
Creation/Data population of XtrChannelInfo_Temp -    Successful
Time taken for XtrChannelInfo_Temp   : 0 s
(8409 row(s) affected)
(8409 row(s) affected)
***Creation/Data population of XtrChannelParameterSpec -    Successful***
Elapsed time: 11 s
```

Purging Service Link Messages

The Service Link Message Purge Utility removes Service Link messages from the database.

Since these messages can be quite large (depending on the complexity of the service form and content type option used to configure the agent), removing the messages greatly reduces the database size required to hold Service Link-related data. External messages remain unchanged.

Purging Messages using the Utility on Oracle Database

To run the utility on Oracle database:

Step 1 Back up the RequestCenter database.

Step 2 Use a query tool appropriate for your database (for example, SQL*Plus) and connect to the RequestCenter database as the RCUser.
Execute the following commands:

SET SERVEROUTPUT ON

EXECUTE sp_CleanupSlMessageContent([FromDate], [ToDate], [UserName]);

Dates must be in DD-MON-YYYY format. UserName is the Service Catalog login name of the person executing the script.

At the end of the execution, the total number of messages purged and elapsed time should be displayed.

Here is an example of the output:

Example:

```
Updating messages with MessageStateID 2 (completed) or 3(failed) that are older than 100 daysDone
updating 3200 messagesScript Start Time 07/06/2011 02:07:11 and script End Time07/06/2011 02:09:11
```

Purging Messages using the Utility on SQL Server Database

To run the utility on SQL Server database:

Step 1 Back up the RequestCenter database.

Step 2 Use a query tool appropriate for your database (for example, SQL Server Management Studio) and connect to the RequestCenter database as the RCUser.
Execute the following command:

EXECUTE sp_PurgeWorkflowTables [FromDate], [ToDate], [UserName]

Dates must be in DD-MON-YYYY format. UserName is the Service Catalog login name of the person executing the script.

At the end of the execution, the total elapsed time should be displayed. Here's an example of the output:

Example:

```
Purge messages with MessageStateID  2 (completed) or 3 (failed)
Done updating 1500 messages
Script Start Time Jul 6 2011 2.57 PM and script End Time  Jul 6 2011 3.57 PM
```

Managing Undelivered Emails

Email notifications that failed to be delivered are kept in the application for review or retry. To delete or resend, navigate to the Administration module, and locate the "Undelivered Emails" tab under Utilities. Delete the messages if they have invalid information, or re-send them if they failed to be delivered due to temporary SMTP outage.

As a good practice, administrator should review this application page on a regular basis to identify messages that need to be re-sent. Email notification process may slow down substantially if there is a large number of messages left in the backlog.

Modifying the First Day of the Week for the Weekly Usage Reports

By default, the weekly usage reports displays the first day of the week as **Monday**. To modify the first day of the week for the weekly usage reports, do the following :

Step 1 Log into your database server as an administrator and run the following query :
Delete from RpWeeklyUsageDetails

Step 2 Edit the **reportsdata.import.beginnerofweek** attribute in the **newscale.properties** file, for example,

Example:

- `reportsdata.import.beginnerofweek= Sunday`

The **newscale.properties** file is located in the **RequestCenter.war/WEB-INF/classes/config** directory.

Managing Different Application Servers

This section provides information about maintaining the application, and managing WebLogic, and JBoss. Additionally, there is information about working with data sources and creating "backing tables" for external data dictionaries, about cached data, application security, applying patches, and multicast settings.

For a typical installation using the JBoss application server, Cisco Prime Service Catalog is started and stopped along with the application server on the command line or Windows services, if they are configured.

Detailed information about starting WebLogic can be found in the Cisco Prime Service Catalog Installation and Upgrade Guide.

The admin server should not need to be restarted during regular Service Catalog operation. There is, however, a need to restart it while installing the custom database driver during installation.

Restarting Cognos Server

The instructions for restarting Cognos applications from the Cognos Configuration Manager or using Windows Services are both Windows-specific tasks, as all Advanced Reporting installations that rely on Cognos components are on Windows systems.

To restart your system:

Step 1 Choose **Start** > **All Programs** > **IBM Cognos 8-64** > **IBM Cognos Configuration**.

Step 2 Choose **Actions** > **Restart**.

Restarting using Windows Services

Stop the following service and then restart:

- IBM Cognos 10.2.1 – required for all reporting options

Deploying the Application

The .war file for Service Catalog is deployed into the file system. The exact location of these files will vary, depending on application server. The Service Link application is provided as a .war file, ISEE.war (Integration Server Enterprise Edition.).

Startup and Shutdown Procedures

This section provides startup and shutdown instructions for the application server, which includes:

- Cisco Prime Service Catalog application
- Cisco Prime Service Catalog Integration Server (Service Link)
- Reporting Server

Restarting Prime Service Catalog, Service Link, and Reporting Servers

Use the Server Console for your application server or command-line scripts as appropriate to restart the server. Ensure make a script available to the Administrator in the development environment.

Key Configuration Files

The following are important files that you may need to see for details on your deployment. Unless specifically stated in this guide or instructed by the Cisco Technical Assistance Center (TAC), all properties files and similar configuration files should be considered read-only. After making changes to any of these property files, you must restart the services.

File	Description
newscale.properties	This file is created by the installer during the install or upgrade process and any time the installer is run. The file produced by the installer is contained in the "RequestCenter.war\WEB-INF\classes\config" folder. As such, the file is redeployed any time the .ear file is redeployed. The Service Catalog administrator should preserve the data contained in the file, but should not restore a copy of the file since the installer may have added new information for the new version. Entries in newscale.properties include: • udk.datasource.jndi – JNDI name for your RC database • udk.datamart.jndi – JNDI name for your data mart database • All registered EJBs • ObjectCache.Application.URL – URL reference back to the application in the emails sent out • ObjectCache.email.host – SMTP host for relaying mail • Container.Datasource – JNDI name for the RequestCenter database • Scheduler.EscalationManagerSchedule – Schedule for evaluating escalations

File	Description
rcjms.properties	This file is also located in the "RequestCenter.war\WEB-INF\classes\config" folder. It contains the JMS settings for application internal communications. Please ensure that the queue names match the ones on the application server.
integrationserver.properties	This file is located in the "ISEE.war\WEB-INF\classes\config" folder. It contains the key properties of the integration server (Service Link).

Managing Logs

Service Catalog maintains log files on the application server to track application activities, both expected and unexpected. Logs are managed using a log4j-based framework, an open source (Apache) logging mechanism. By default, logs are configured as "rolling appenders", with a new log file opened every day. Location of the log files varies according to the application server type, as does the ability to adjust log file contents and configuration.

We recommend that you:

- Rotate logs on a daily basis (this is the default behavior)
- Keep one month of logs "online"
- Back up or delete logs that are older than a company-specified retention period

Service Catalog does not require log files to be maintained. They are useful primarily as troubleshooting tools in case an error arises. There are four types of log entries: **E** (Error), **W** (Warning), **I** (Info), **D** (Debug), listed in decreasing order of severity.

We recommend against changing the format of the default log files because that is the format the Cisco Technical Assistance Center (TAC) expects. Rather, customers can create their own appenders that suit their needs.

In addition to the system-wide log files, Service Link is configured to have a separate log file for each adapter type. These logs, too, are managed by log4j. By default, Service Link logging is enabled. The adapter-specific log files are written to the ServiceLink\logs directory:

- dbadapter.log
- fileadapter.log
- httpadapter.log
- msadapter.log
- mqadapter.log
- remedyadapter.log
- vmwareadapter.log
- wslisteneradapter.log

With full DEBUG level enabled, logs get very large very quickly, so logging at full debug and trace levels should be enabled only for short periods. System performance will likely slow down significantly, so logging on a Production system should be kept to a minimum, and only for the length of time required to reproduce an issue. For more information, see Logs and Properties.

WebLogic Logging

In WebLogic, Cisco Prime Service Catalog routes messages according to the WebLogic logging configuration.By default, all logging goes to the WebLogic server log, which is usually found in a path similar to the following:

/apps/bea/user_projects/domains/cisco/servers/nsServer/logs/nsServer.log

The default log level is set to INFO, and is adjustable via the WLS Console.

JBoss Logging

The JBoss logs are located under "<JBOSS_DIR>/standalone/log" folder. The logging.properties file that determines logging behavior is located under the "<JBOSS_DIR>\standalone\configuration" folder. Log4j.xml is no longer used for controlling application logging.

Configuring Data Sources

All modules depend on J2EE data sources, defined via JNDI (Java Naming and Directory Interface). These data sources must point to the correct database and have the appropriate login information configured.

Additional JNDI data sources are required if:

- External dictionaries are used.
- Customer-specific data sources are accessed by data retrieval rules or by option lists in a service definition that are based on a SQL statement or a relational database table.

Accessing external data sources on a type of database different than Service Catalog (for example, a SQLServer data source accessed from an instance of Service Catalog running on Oracle, or a Sybase data source accessed from any instance of Service Catalog) is not supported in a service form. Procedures for configuring data sources are detailed in the Cisco Prime Service Catalog Installation Guide, and are specific to the application server.

When you add data sources, you should use the Cisco drivers if possible.

You can configure a custom data source using JBOSS 7.1.1.

Step 1 Log on to the JMX admin console of JBOSS Application Server.

Step 2 Choose **Datasources**.

Step 3 Select **Validation** tab.

Step 4 Enter the following data in the **Valid Connection Checker** field.

- Select 1 (for SQL server)
- Select 1 from dual; (for Oracle server)

Note You must populate and enable Valid Connection Checker while configuring Datasources to prevent intermittent SQL disconnections.

Step 5 Uncheck the **Validate on Match** check box to make it false.

Step 6 Check the **Background Validation** check box to make it true.

Step 7 Enter **Validation Millis** as 600000.

Step 8 Click **Save**.

Step 9 Select **Pool** tab.

Step 10 Enter the **Min Pool Size** as 1 and **Max Pool Size** as 10.

Step 11 Click **Save**.

Step 12 Select **Properties** tab.

Step 13 Enter the **Strict Minimum** and **Prefill Enabled** values as True.

Step 14 Click **Save**.

Creating Backing Tables for External Dictionaries

External dictionaries within the Service Catalog need to be backed by physical tables in the database. You cannot have read-only external dictionaries. All external dictionaries are read-write. Only the application should write to External Dictionaries.

For the application to relate External Dictionaries to the Requisition, a numeric column needs to be available that can be used as the foreign key. This is typically named RequisitionEntryID.

Sample SQL Listing to Create a Backing Table

This code creates a sequence that generates unique ids for each row. Creating an index on the RequisitionEntryID column greatly optimizes Service Manager performance.

The backing tables for external dictionaries are not transported by Catalog Deployer across environments. Only the dictionary definition can be deployed, as a component of a service.create sequence X_SEQ;

```
create table (
  X_ID INT CONSTRAINT PK_X primary key,
  REQUISITION_ENTRY_ID INT,
  REQUESTORLANID VARCHAR2 (10),
  REQUESTORNAME VARCHAR2 (50),
  FUNDINGSOURCECODE VARCHAR2 (15),
  DATENEEDED DATE,
  REASONFORCHANGE VARCHAR2 (50),
  PROJECTNAME VARCHAR2 (50),
  TOPINITIATIVE VARCHAR2 (5));
create or replace trigger X_it
  before insert on X for each row
declare
  seq_val number;
begin
  select X_seq.nextval into seq_val from dual;
  :new.X_ID := seq_val;
end;
```

Configuring Service Export via SSL or NTLM

The Service Export feature in Service Designer establishes a connection to Service Catalog, retrieves the exported XML, stores it in a file, and returns a link to the user.

If the application is SSL-enabled, then the user will encounter a problem when trying to export a service as an XML document. The connection to the application needs to authenticate to the server, and the Service Catalog needs an SSL certificate.

To enable the export service feature when Service Catalog is SSL-enabled:

Step 1 Export the trusted root CA certificate used by the Service Catalog web server, in Base 64 Encoding format, into a file. The file will have an .arm or .cert extension. This is a simple text file that can be opened in any text editor.

Step 2 Find the CA certs keystore that comes with the Java installation on your application server. The CA certs keystore for your Java installation is a file named cacerts.

- For JBoss, cacerts is located in <JAVA_HOME>\jre\lib\security.

Step 3 Import the trusted root CA certificate of the Service Catalog web server into the Java's cacerts keystore. You can also use the Java keytool utility.

- The keytool.exe program can be found in the <JAVA_HOME>/bin directory.

The following example provides the command line syntax for the Java keytool utility, which will import the root CA certificate into cacerts:

Example:

```
keytool.exe -import -trustcacerts -alias RC –file <root_cert_file> -keystore
C:\jdk1.6.0_12\jre\lib\security\cacerts
```

where <root_cert_file> is the full pathname of the file that contains the root CA certificate of the Service Catalog web server which you exported in step 1. The keytool program will prompt you for a keystore password. For a new installation of Java, the default keystore password for the **cacerts** file is **changeit**. Enter **changeit**, or another value if you have already changed the password since you installed Java on this machine. If the question **Trust this certificate?** appears, enter **y**.

Step 4 Restart the application server instance, for the changes to take effect. Restart the whole instance of JBoss WebLogic in this machine, and not just an individual server or application.

Reloading Cached Data Settings

Most site configuration settings are cached in the J2EE system for faster access. To reload any settings that are used by the J2EE application, change any option on the Settings page of the Administration module and click **Update**. This invalidates the cache and reloads the settings from that page.

Business Engine Caching

Cisco Prime Service Catalog includes a proprietary work flow management system, sometimes referred to as the Business Engine. The actions of the Business Engine—managing the delivery plan—are largely transparent to application users, since they occur on the application server. However, a user interface is provided for system administrators to view and possibly adjust Business Engine operation.

Users with the Site Administrator system role can access the Business Engine console via the URL http://<serverName:portNumber>/RequestCenter/businessengine/index.jsp, where you can:

- View the Business Engine configuration
- Delete the Object Cache
- Force a run of the Escalation Manager
- View the transaction cache log

Other caching mechanisms are also in place within the application. The cached values are refreshed automatically as and when changes are made to the application data.

Securing Prime Service Catalog Database

User passwords are usually not stored in the database if external authentication via SSO is used. When they are, they are a one-way AES 128-bit hash. Passwords stored in configuration files or in the database are encrypted using a Public/Private key encryption. No additional encryption is applied to the data.

Local application passwords in configuration files are encrypted. When Service Link is configured, the J2EE container password is not encrypted and is stored as plain text in several configuration files.

URLs are not encoded; data-level security verifies authorization for each screen.

Securing Application

This section describes about the application security:

- Retrieve SSL certificates from the LDAP server.
- Ensure LDAP server supports SLDAP connectivity (typically on port 636).

Service Catalog maintains a password-protected key-store that can store many certificates.

We recommend that the web server, or the content switch in front of the web server run SSL, especially, in Extranet-supported environments.

Web Server to Application Server communication does not usually need to be encrypted.

Removing CGI support in Advanced Reporting

Several tools scan applications to ensure that no CGI-based submits (GET Form submissions) exist in the application.

Cross-Site Scripting

Cisco is focused on the security and safety of your data and is well aware of the threats presented by XSS (Cross-site scripting) attacks.

Service Catalog uses a standard J2EE **input-filter-config.xml** file to check that URLs do not contain any of the following characters: < > " ' ( ) & ;

This file is located in: RequestCenter.war\WEB-INF\config\.

Form Data Security

For installations that are on release 9.3.2 and later, there are a number of service design features that can be used to protect service requests from the malicious files. To prevent malicious attempts to intercept form data that are governed by form rules and default value settings, server-side rules and certain edit controls can be used to override or validate data being sent from the browser clients. For more information, see Cisco Prime Service Catalog Designer Guide.

Reporting Batch Programs

The Reporting modules require scripts that maintain the Service Catalog data mart and produce the standard reports and KPIs available to users.

Service Catalog Extract-Transform-Load (ETL) scripts generated from the Cognos DataManager ETL tool controls the population of the database which supports running prebuilt reports provided by Service Catalog and all nonform based data in the data mart.

Additional command files complete the generation of the framework used by Cognos QueryStudio and Report Studio (Ad-Hoc Reports and Report Designer) to permit ad-hoc reporting on the Service Catalog data mart.

These scripts share the same invocation and logging framework. They are available as Windows .cmd files that reside and run on the Cognos server. They can be scheduled to run via any enterprise scheduler. These scripts log their activities in the <ReportingInstalledDirectory>\logs directory of the Cognos server.

The following script is required to support standard reports and Key Performance Indicators (KPIs).

Table 6: Support standard report

Program	Description/Usage
update_datamart_std.cmd	Populates database tables which support the prebuilt reports according to ETL rules specified in Data Manager. This is a complete rebuild of the database contents, rather than an incremental refresh. Creates a log file in *<cognos.root >\c8 \datamanager* \log.

The following programs are required to support the data marts.

Table 7: Support data mart

Program	Description/Usage
update_datamart.cmd	Populates the data mart fact and dimension tables using rules specified in Data Manager, as well as the Demand Center data mart. This is an incremental refresh of all static dimensional and fact data. It creates a log file in *<cognos.root >\c8 \datamanager* \log.
create_model.cmd	Creates a Cognos FrameworkManager model that includes dynamically defined reportable objects (dictionaries and services) as well as standard facts and dimensions. The model is rebuilt by merging a statically defined model (the standard facts and dimensions used in the data marts) with dynamically generated metadata describing reportable services and dictionaries.
publish_fdr_pkg.cmd	Publishes the FrameworkManager model to the Cognos BI Server, via the Cognos ScriptPlayer utility. Must be run as part of the Service Catalog data mart refresh, following the program that creates the model (create_model.cmd).

Form-Data Extraction Script

Dictionaries and services designated as reportable are populated in the data mart by a Java program. The program activities are logged in the current log file on the application server.

This program is run via the internal scheduler. Schedule settings can be specified as part of the installation or modified by editing the newscale.properties file. The following properties configure the scheduler. We recommend running the ETL (and other processes) daily. The data mart will not be usable when the job is running. The ETL process is run with transaction logging. It may be advisable to increase the transaction size (FDR_ETL_RECORDS_PER_BATCH).

```
#Enable ETL Process: 0 or 1 (1=Yes, 0=No)
ENABLE_FDR_ETL_PROCESS=0
# FDR_ETL_TRIGGER : 1 for hourly, 2 for daily, 3 for minutes
FDR_ETL_TRIGGER=1
#Frequency Hourly
FDR_ETL_TRIGGER_FREQUENCY_HOURLY=5
#Daily Time HH:MM (22:30 for 10:30 PM)
FDR_ETL_TRIGGER_FREQUENCY_DAILY=22:30
#Frequency in minutes
FDR_ETL_TRIGGER_FREQUENCY_MINUTES=1
#Number of records per batch insertion
FDR_ETL_RECORDS_PER_BATCH=500
```

Monitoring Tasks using Escalation Manager

The Escalation Manager is responsible for monitoring if a task exceeds its Operating Level Agreement (OLA). If the OLA is exceeded, and escalations have been configured, the Escalation Manager sends the appropriate notifications after the designated amount of time has elapsed since the task became overdue.

The Escalation Manager is run via the internal scheduler. Schedule settings can be adjusted by editing the newscale.properties file. By default the Escalation Manager is set to run during business hours Monday through Friday.

A schedule setting is essentially a cron expression, which describes the desired schedule in the format “Seconds Minutes Hours Day-of-Month Month Day-of-Week”. For example, the expression “0 0 12 ? * WED” means “every Wednesday at 12:00 pm”.

Fulfilling Service Requests using Service Manager

Service Manager is the module used by task performers to fulfill service requests.

Service Manager allows users to search for tasks or requisitions of interest by specifying a set of conditions to be matched, via the Filter and Search pop-up window. By default, these conditions do not support a **Contains** operator, for example, the ability to find all task whose name contains a specified string.

This default behavior optimizes performance by increasing the probability that indexed queries can be run against the database. The functionality of performing **Contains** queries can be supported; however, administrators should be careful in making this configuration change, as response time may not be optimal, especially with a large transactional database. Reverting is not recommended as it will impact Service Manager Custom Views.

To allow Service Manager users to specify **Contains** queries, edit the newscale.properties file, to add the following property setting:

```
# Service Manager will use this flag to control Contains Query in Datatable Filter and
Search
ContainsQueryInFnS=true
```

For the changes to take effect for the newscale.properties files, navigate to the **Administration > Utilities** modules, select “Request Center – Property Files” from the drop down, select “newscale.properties” file and click the “View File” button. Review the file content to make sure it includes your changes, and then click the “Reload” button. Click “OK” button when the reload success message is displayed.

Installation Log Files

Installation Logs are saved to the <APP_HOME>/logs folder with a mmddyyyyhhmm time-stamp (for example, 010720111126) each time the Installer is invoked.

Key installation logs are listed below.

Table 8: Installation log files

File Name	Contents
RC_Install	General installation logs.
RC_File	Information about any files that were added, moved, or deleted from the file system.
RC_DbInstall	Information about the SQL scripts executed during the database installation/upgrade process, including the time taken for each script to execute.

File Name	Contents
RC_Sql	Log of the SQL statements that were run on the database during the install. This log may be particularly useful if a SQL script fails during the installation, as the log will contain the text of the script which caused the error and indicate the exact nature of the error.

Installation settings are recorded in the RequestCenter/etc folder. Preserve that folder so that installation settings are stored for future invocations of the Service Catalog Installer.

The settings are available in the file setup_options.txt.

Multicast Settings

A single clustered installation of Cisco Prime Service Catalog requires multicast to communicate within the cluster. Each node has to be on the same subnet or have multicast routing enabled across the subnets on the switches. You may also have to enable multicasting in the network interface configuration of the host servers.

Service Catalog uses multiple multicast addresses that have to be unique.

Testing Multicast Connectivity

This section describes how to test multicast connection. You can perform a test to check if Node1 can talk to Node2, as follows:

1. Choose a valid multicast address and port that are not in use.
2. On Node2: java -classpath .:./javagroups-all.jar org.javagroups.tests.McastReceiverTest -mcast_addr 224.10.10.10 -port 5555.
3. On Node1: java -classpath .:./javagroups-all.jar org.javagroups.tests.McastSenderTest -mcast_addr 224.10.10.10 -port 5555.
4. On Node1 you see a prompt “>”.
5. Type in some text and press Enter.
6. Your text appears on Node2.

You can also check if Node2 can talk to Node1:

Repeat the procedure (test) stated above with Node2 as the Sender and Node1 as Receiver.

For managing integration, the key integration strategies that the system administrator must pursue when configuring the Service Catalog application are described in Cisco Prime Service Catalog Integration Guide.

Directory Integration

The system allows for multiple LDAP directory integrations. A group of two or more LDAP sources becomes one LDAP system through referrals. Referrals are supported for searches only, not binding. For detailed information on configuring directory integrations, see the Cisco Prime Service Catalog Integration Guide.

Directory Integration allows integration architects to connect Service Catalog to an LDAP data source and map attributes in that data source to corresponding fields in the Person profile. The integration allows designers

to designate which events should trigger an LDAP lookup, and whether that lookup should also cause a refresh of the Person profile in Service Catalog. Events that can trigger an LDAP lookup include:

- Authentication after login, either via the Service Catalog screen or Single Sign-On
- Person Search for Order On Behalf
- Person Search for form data in a Person-type field
- Lookup of Person information for the managers of a person previously chosen via Order on Behalf or Person Search

In addition to these preconfigured events and behavior, Directory Integration provides an API to allow programmers to implement custom directory interfaces to add new search capabilities or refine the search logic.

Directory Mappings

Directory data can be mapped to elements of a Person's profile including:

- Basic and extended person attributes, including location and contact information
- One or more organizations
- One or more groups
- One or more roles

Four types of mappings are available:

- Simple mapping. A 1-to-1 mapping between a directory attribute and a Person field.
- Composite mapping. Two or more directory attributes are used to derive the value of a Person field.
- Expression mapping. A regular expression involving one or more directory attributes is used to conditionally derive the value of a Person field.
- Mapping via Java class, using the Directory Integration API. A Java plug-in derives the value of the Person field based on directory attributes available in the current directory data source for the current person.

If the Locale and Time Zone are not mapped, Service Catalog uses the server default. Also, if any optional fields are not mapped, any value previously populated in the Person profile remains unchanged.

Custom Mappings

Custom mappings can be created via pattern-matching language (regular expressions), which is described in the Cisco Prime Service Catalog Integration Guide , and via a custom plug-in class based on an interface provided in the Directory Integration API.

Any such mappings should be documented in the LDAP Integration document for each implementation. Any Java classes required for the mapping are treated as customizations if/when a Service Catalog instance is migrated or upgraded.

Custom Code

Using the interfaces provided by the Directory Integration API, custom Java classes can replace or supplement the preconfigured behavior offered by the directory integration events. Any such classes are treated as customizations when/if an instance is migrated or upgraded.

Further, if the custom classes require supporting JAR files, these must be installed on the application server and treated as customizations. Installation procedures differ for each application server.

Troubleshooting Single Sign-On

Single Sign-On functionality is provided as part of Directory Integration. If you experience any problems with Single Sign-On, begin troubleshooting by checking the following items:

- Review any related changes to your environment such as LDAP or Junction/SiteMinder agent configurations.
- Check if the Service Catalog is still accessible through the Administrative override
- Restart the Service Catalog service.

Single Sign-On: Configuring NTLM

Many environments use Windows authentication. IIS supports Integrated Windows Authentication (IWA) and passes the DOMAIN\UserName of the user who is logged in as a parameter.

Requirements

- Restart the IIS Admin Service (in Windows Services) after enabling IWA
- Valid domain accounts while accessing Service Catalog
- Configure SSO to strip DOMAIN information

Interactive Service Forms (ISF)

ISF is a JavaScript API that integrates with Cisco Prime Service Catalog service forms. ISF allows the forms to dynamically alter their contents or behavior based on the current context, including user credentials; data previously entered on the form; or the life cycle of the displayed requisition. For more information on ISF, see the Cisco Prime Service Catalog Designer Guide.

ISF supports the use of JavaScript libraries, stored on the application or web server, to supplement JavaScript code stored in the Service Catalog repository. If such libraries are used, they are treated as customizations when upgrading or migrating a Service Catalog site.

Retrieving Data using Active Form Components

The data retrieval rules available within active form components allow the application to retrieve data from external relational databases or from the application database, for use in service forms. Such data can be used to prefill form fields with default values; to produce drop-down lists; and to provide dynamically populated drill downs to detailed information. User data entry could also be validated against the external data.

For a rule to access an external database, a corresponding JEE datasource must be created. Instructions on creating the datasource are given in the Cisco Prime Service Catalog Installation and Upgrade Guide. Any such datasources are treated as customizations when upgrading or migrating the Service Catalog site.

Integrating with External Systems using Service Link

Service Link, also known as the Integration Server, or ISEE (Integration Server Enterprise Edition), allows Service Catalog to send synchronous or asynchronous requests to other systems via XML messages. Tasks that are configured in Service Designer as “external” are handled by Service Link.

Service Link uses JMS queues as an underlying technology, so disruption to JMS configuration may disrupt Service Link operation. Most Service Link troubleshooting can be done through the Service Link module which provides the ability to drill-down to individual messages sent or received and the tasks responsible for sending or receiving those messages.

Including Custom Content during Installation

This section provides information about configuring your system for a customized installation of Service Catalog, and ensuring that custom content is not deleted or overridden during subsequent installations or upgrades.

For more details on the Service Catalog installation wizard, see the Cisco Prime Service Catalog Installation and Upgrade Guide.

How the Installer Works

The Service Catalog installation wizard builds the WARs and:

- Expands the core product WAR
- Modifies .properties files based on settings chosen during installation
- Merges in a customizations file, if one is specified as part of the installation parameters
- Rejars the WAR
- Publishes the WAR to the dist/folder for deployment

The deployment procedure stipulates that an entire WAR file be deployed to a server. When an entire WAR file is deployed, the previous directory where the WAR was expanded is wiped clean, and any Service Catalog customizations that existed in the directory are lost.

To avoid losing the customizations, the Service Catalog installation wizard allows you to specify custom content to be included in the installation:

Procedure

Step 1 Create an archive containing the custom content in the Zip format. The archive directory structure must match the deployment directory structure.

Step 2 Run the Service Catalog installation wizard as described in the Cisco Prime Service Catalog Installation and Upgrade Guide, using the **Advanced Installation** type.

Step 3 On the Application Server Configuration page, click **Advanced Options**.

Step 4 In the The Advanced Options dialog box, select **Custom content**.

Step 5 Enter the full path in the **Custom content archive** including the name of the archive, or click **Browse** to locate and choose the custom content archive.

Step 6 Click **Close**. Continue with the installation as described in the Cisco Prime Service Catalog Installation and Upgrade Guide.

While the Service Catalog installation wizard completes the installation, it extracts your custom content archive into the application deployment directory structure.

Implementation-wide Custom Files

All customized files should be included in the customization archive. The following customized files may be required at all sites within an implementation:

Table 9: Custom components

Customizable Component	Directory/Files
Custom style sheets, headers, footers	RequestCenter.war\custom*\custom.css RequestCenter.war\custom*\portal-custom-header.css RequestCenter.war\custom*\images\ RequestCenter.war\custom*\header.html, footer.html, for all directories on which custom style sheets have been installed
ISF libraries	RequestCenter.war\isfcode*
Custom Classes	RequestCenter.war\WEB-INF\classes\ (custom classes such as those related to Directory Integration customization)
Property Files edited by hand (such changes could also be site-specific)	newscale.properties rcjms.properties integrationserver.properties newscalelog.properties

Database Scripts

We do not recommend modifying the database outside of the APIs provided by Cisco. However, some scripts may need execution directly against the database.

External Dictionaries

External Dictionaries are stored as database tables. Whenever these dictionaries are modified, DDL scripts need to be run to modify the corresponding table.

Patches

Customer Support may provide a SQL script as part of a patch or hotfix that needs to be run manually. Until a hotfix is included in a subsequent product release, it must be treated as a customization to be included in a software upgrade or reinstall.

Managing Configuration using Catalog Deployer

A Service Catalog implementation typically consists of multiple sites, each of which plays a different role:

Table 10: Multiple sites

Site	Usage
Development	Service definitions are developed and unit tested; customizations are initially applied
Test	A controlled environment, not interrupted by development activities, where Quality Assurance or other personnel test a Service Catalog
Production	The live environment where the user community can request services from the Service Catalog and IT teams can fulfill service requests

The Catalog Deployer module provides configuration management for metadata (service definitions) and organizational data (people, organizations, and related entities) which is stored in the repository. For more information, see Managing Content Deployment for Catalog Deployer documentation.

Copying a Database

You can copy Service Catalog OLTP database from one site to another during deployment, for instance:

- When initially installing a test or production site, the complete development site may be copied to the new environments.
- After production has been in operation for a time, all of the user activity should be copied to a test environment, to allow realistic performance or volume studies.

Perform the following procedures to copy a Service Catalog OLTP database from one site to another.

Exporting Source Database

Step 1 Inform the users of expected downtime.

Step 2 Stop the Service Catalog and Service Link services in the source environment.

Step 3 Export the source database. Develop a naming convention that allows you to track the source of the data and the date of the export.

Step 4 If a system shutdown is not feasible, use the –consistent flag for the Oracle export.

Step 5 Restart the Service Catalog and Service Link services.

Importing Database to Target Site

Step 1 Stop the Service Catalog and Service Link services in the target environment.

Step 2 Ensure you have a current backup copy of the target database.

Step 3 If required, copy the export file from its destination to a file system accessible to the target database server.

Step 4 Import data into the target database.

Note For SQLServer, ensure that logins and users exist in the newly imported database match the credentials required for this instance of Service Catalog. If required, create a new login or associate an existing login with the database owner and ensure this user has appropriate permissions. For Oracle, ensure appropriate users exist in the newly imported database with privileges as specified in the Service Catalog installer.

Step 5 If the two sites are accessing two different Cognos reporting servers, update the entry in the CnfParams table that specifies the name of the “CognosServer” for this site and commit the update.

Step 6 Restart the Service Catalog and Service Link services in the target environment.

Step 7 Set the**Administration** > **Entity Homes** > **SiteProtection This Site Is**property to the current site. If Entity Homes are specified differently, or sites have different protection levels, make the changes manually and save your changes.

Step 8 If the two sites are connecting to two different LDAP directories, adjust the Directory Integration Data Source definition appropriately.

Step 9 Check and modify any connection properties for the Service Link agents as appropriate for the target environment.

Step 10 Perform any additional manual operations to adjust the data. For example, you may wish to add permissions to some people, groups, or organizations, or revoke permissions.

Step 11 Inform users that the maintenance is complete.

Configuring SSL for Service Link Inbound Documents

This section describes about configuring SSL for service links.

Enabling SSL for the Service Link service involves:

- Getting a digital certificate that is either self-signed or signed by a known CA such as VeriSign.

- Installing the certificate, and
- Configuring a secure port number for the application server on which the Service Link service is running.

Procuring a certificate signed by a well-known Certificate Authority like VeriSign or Thawte has the benefit that most client programs already recognize the signer certificate from one of these Certificate Authorities.

If you choose to use a self-signed certificate for your Service Link service, then you must exchange the signer certificate with all external systems that communicate with Service Link via web interface.

For example, if an external system sends a response message to a Service Link agent which uses the http/ws adapter for its inbound adapter, then that external system acts as a client that connects to Service Link via an **https** URL, and will need to understand how to complete the trusted handshake for a successful SSL connection.

In order to do this, the external system needs to recognize the signer for the certificate used by the Service Link service. To achieve this, the signer certificate for Service Link must be imported into the *Trusted Certificate Authority Keystore* of the external system. More detailed instructions are given later in this section.

Note Service Link, as a server, does not support client certificate authentication during SSL handshake.

Enabling SSL for Service Link

Enabling SSL for Service Link turns on the secure port, but it does not turn off the nonsecure port for Service Link. If you choose not to turn off the nonsecure port, external systems can still communicate with Service Link via an http URL. If you decide to turn off the nonsecure port, all communications with the Service Link service must use the **https** URL.

It is possible to use both secure and nonsecure port for the Service Link service and control the access to the nonsecure port via another mechanism, such as a firewall system.

For example, in a Two-JBoss-Server topology, the Service Catalog application is also a “client” of the Service Link service (which runs on a separate JBoss server). At runtime, Service Catalog needs to connect to the Service Link service via the URL http://<*SL_servername* >:6080. If the nonsecure port 6080 is turned off for the Service Link service, then Service Catalog must be configured to connect to Service Link via an https address, that is, **https**://<*SL_servername* >:**6443**.

So, one possible scenario is that you turn on both nonsecure port 6080 and secure port 6443 for the Service Link service. Service Catalog can still connect to Service Link via http://<*SL_servername* >:6080, while other external systems must only communicate with Service Link via https://<*SL_servername* >:6443. You configure your firewall system to deny access to port 6080 from all external systems.

This section does *NOT* describe how to turn off a nonsecure port for the application server or how to configure a firewall system to deny access to a nonsecure port number. Please contact your system administrator, or the vendor of your application server product to obtain the information you need.

If Service Link is deployed in a separate application server from Service Catalog (as in the case of a clustered WebLogic environment, or in the case of a Two-JBoss-Server topology), then to enable SSL for Service Link, you configure the certificate and secure port number only for the application server where Service Link is running.

Creating a Certificate Keystore

It is assumed that you have procured a digital certificate that can be used to secure the Service Link service. This certificate can either be self-signed or obtained through a third-party Certificate Authority like VeriSign. In either case, your digital certificate must be imported into a java keystore (that is, a jks file) that can be accessed by the application server. Furthermore, the signer certificate (aka the public key of your certificate) must be exported into a file in "Base64-encoded ASCII" format, so that it can be given to the external systems that want to communicate with Service Link service in SSL mode.

Note This document does not describe how to create a keystore file, and how to request a certificate for your web server or application server. The instructions in this section assume that you have already created a keystore file that contains the digital certificate to be used to enable SSL for the application server where Service Link is running. For ease of documentation, assume that your keystore file is named "**slkeystore.jks**". It contains a certificate under the alias called "**servicelink**". The password to open this keystore file is "**slpassword**".

Also assume that the signer certificate has been exported in "Base64-encoded ASCII" format into a file named "**slsigner.ce**r". A "Base64-encoded ASCII" format is similar to the following example:

```
-----BEGIN CERTIFICATE-----
MIICPDCCAaUCBE17w1cwDQYJKoZIhvcNAQEEBQAwZTELMAkGA1UEBhMCVVMxCzAJBgNVBAgTAkNB
MRIwEAYDVQQHEwlTYW4gTWF0ZW8xETAPBgNVBAoTCG5ld1NjYWxlMQswCQYDVQQLEwJRQTEVMBMG
A1UEAxMMS2hhbmcgTmd1eWVuMB4XDTExMDMxMjE5MDI0N1oXDTIxMDMwOTE5MDI0N1owZTELMAkG
A1UEBhMCVVMxCzAJBgNVBAgTAkNBMRIwEAYDVQQHEwlTYW4gTWF0ZW8xETAPBgNVBAoTCG5ld1Nj
YWxlMQswCQYDVQQLEwJRQTEVMBMGA1UEAxMMS2hhbmcgTmd1eWVuMIGfMA0GCSqGSIb3DQEBAQUA
A4GNADCBiQKBgQDhTxg2RwarD6Wn4iqYeOOk3ykfXzZiDArf/X63omXquTmN0Up+mg6oJmPAfqJA
l7k4+Dn7dfVtAc4h8qra7PBeBU48zrzRqZd6VAK07rz++CilQto64mHXYVomb5vWPGeKA41j9vlv
ENj/tE/6++IqbwnxAqeZtY3EvEM7dcCWDwIDAQABMA0GCSqGSIb3DQEBBAUAA4GBAAaqCnfEAovy
Uf2S+oAXYDo5N387a035APsz5iiUM5oiKR/KW3oRz/v0P0I/o3n312kDIJ01111pl6qpZRtPEsr1
b00Tu1cXfPmizEtz0ole606qDS+DzkS1+YYz2mLL2Zq40d1EPsMolyqyUmyq3GHaEnuhWemcv2aA
wGFgbQYd
-----END CERTIFICATE-----
```

Skipping Certificate Validation

You could choose to skip the certificate validation for an SSL connection by checking the **Skip Certificate Validation** option when you create a connection. When this option is selected the Certificate validation is skipped and the connection is established without the Certificate Keystore information.

Installing the Keystore for the Application Server

The subsequent sections contain instructions for installing the certificate file and configuring SSL for each type of application server.

For JBoss 7.1.1

Step 1 Stop the JBoss server where the Service Link application is running.

Step 2 Copy the "**slkeystore.jks**" file into the "<JBOSS_DIR>\ServiceLinkServer\configuration" directory, where <JBOSS_DIR> is the installation directory of the JBoss server where the Service Link application is deployed.

Step 3 Make a back up of file "<JBOSS_DIR>\ServiceLinkServer\configuration\standalone-full.xml". Use a text editor to open file "standalone-full.xml". Ensure you use a text editor that will not insert any special carriage return characters or any other formatting characters into the file.

Step 4 Search for the following line in file "standalone-full.xml":

Example:

```
<connector name="http" protocol="HTTP/1.1" scheme="http" socket-binding="http"/>
```

Insert the following three lines right below it:

Example:

```
<connector name="https" protocol="HTTP/1.1" scheme="https" socket-binding="https" secure="true">
<ssl name="ssl" key-alias="servicelink" password="slpassword"
certificate-key-file="../ServiceLinkServer/configuration/slkeystore.jks"/>
</connector>
```

Note In the above entries, it is assumed that the name of your keystore file is "slkeystore.jks", the alias for the certificate is "servicelink", and the password to open the keystore file is "slpassword".

Search for the following string in file "standalone-full.xml":

Example:

```
<socket-binding name="https" port=
```

Step 5 Make a note of the value for port number. This will be the secure port number used by the JBoss server in SSL mode.

Step 6 Stop the JBoss server where the Service Catalog application is running, and Navigate to the "<JBOSS_DIR>\ServiceCatalogServer\deployments\RequestCenter.war\WEB-INF\classes\config" directory.

Step 7 Use a text editor to open file "newscale.properties", and search for the following parameter:

Example:

```
isee.base.url=
```

Note The Service Catalog application is communicating with the Service Link application via this URL. This Service Link URL is now SSL enabled, and thus the address needs to be changed to an https address, and the port number needs to be changed to the secure port number used by the JBoss server for Service Link.

Step 8 Change the value for this parameter from http://<hostname>:<nonsecure_port_number> to **https**://<hostname>:<**secure_port_number**>.

Step 9 Copy file "slsigner.cer" to the "<JAVA_HOME>\jre\lib\security" directory, where <JAVA_HOME> is the JDK 6 installation directory.
It is assumed that file "slsigner.cer" contains the CA certificate.

Step 10 Open a Command Prompt window or a Console window and navigate to the "<JAVA_HOME>\jre\lib\security" directory.

Step 11 Execute the following command to import the CA root certification into the trusted certificate keystore used by JDK 6:

Example:

```
<JAVA_HOME>\bin\keytool -import -trustcacerts -file slsigner.cer -alias servicelink -keystore cacerts
 -storepass changeit
```

Note In the above entries, it is assumed that "slsigner.cer" is the name of the file that contains the CA root certificate, "servicelink" is the alias, "cacerts" is the name of the trusted keystore file for JDK 6, and "changeit" is the password to open the "cacerts" keystore file.

Step 12 Start both ServiceCatalogServer and ServiceLinkServers, and connect to the Service Catalog URL as an administrator user, or as a user who can access the Service Link module.

Step 13 Open the Service Link home page, in the Service Link Status section, verify that the connection is in green status, and both the SSL icon and the secure port number are displayed.

Step 14 Any external system that sends an inbound document to the Service Link agent that uses the HTTP/WS adapter will need to be updated as follows:

a) The inbound routing URL needs to use the https address and the secure port number.

b) The signer certificate for Service Link (contained in file slsigner.cer) will need to be imported into the trusted CA root certificate keystore of the external system.

For WebLogic 10.3.6

Perform the following steps as a user who can access the WebLogic Administration Console:

Step 1 Copy the certificate keystore file "slkeystore.jks" to the "<*JAVA_HOME* >\jre\lib\ security" directory on the WebLogic machine where Service Link is running.

Note In a clustered WebLogic environment, Service Link must be deployed in a WebLogic server that does not belong to the cluster. So, make sure you find the correct WebLogic server for Service Link.

Verify that <JAVA_HOME> is the correct Java directory used by the WebLogic application server. Look for the JAVA_HOME setting inside file "commEnv.cmd" (on UNIX/Linux, look for "commEnv.sh"), located under the "<WL_HOME>\common\bin" directory. For example: set JAVA_HOME= C:\Program Files\Java\jdk1.7.0\

Step 2 Log on to the WebLogic Administration Console and navigate to***<<domain>>*** **Environment > Servers**.

Step 3 Click the name of the WebLogic server for Service Link to open its configuration settings, and click the **Configuration >Keystores** subtab.

Step 4 On the Keystores page, enter the following values.

Note Replace <*JAVA_HOME* > with the full pathname of the Java Directory. (For the read-only fields, verify the values that appear are correct.)

Table 11: Keystore fields

Field	Value
Keystores	Custom Identity and Java Standard Trust
Custom Identity Keystore	<*JAVA_HOME* >\lib\security\slkeystore
Custom Identity Keystore Type	jks

Field	Value
Custom Identity Keystore Passphrase	slpassword
Confirm Custom Identity Keystore Passphrase	slpassword
Java Standard Trust Keystore	<*JAVA_HOME* >\lib\security\cacerts
Java Standard Trust Keystore Type	jks
Java Standard Trust Keystore Passphrase	changeit
Confirm Java Standard Trust Keystore Passphrase	changeit

For the Java Standard Trust Keystore Passphrase: It is assumed that the password for the "cacerts" keystore file is still the default value of "changeit". Replace it with the correct value if the password for "cacerts" has been changed in your environment.

Step 5 Click **Save**, and click the **Configuration > SSL** subtab.

Step 6 On the SSL page, enter the following values:

Table 12: SSL fields

Field	Value
Identity and Trust Locations	Keystores
Private Key Alias	servicelink
Private Key Passphrase	slpassword
Confirm Private Key Passphrase	slpassword

Step 7 Click **Save**, and click the **Configuration > General** subtab.

Step 8 On the General page, enter the following values:

- Check the **SSL Listen Port Enabled** check box.
- SSL Listen Port = <*enter an available port number, for example 9443* >.

Step 9 Click **Save**, restart the WebLogic server where Service Link is deployed.

Step 10 Check if the log file "<*WL_servername* >.out" contains messages similar to the following, to ensure that the WebLogic server has started in the secure port (9443):

Example:

```
<Notice> <Security> <BEA-090171> <Loading the identity certificate and private key stored under the
 alias hydra2 from the jks keystore file C:\jdk160_23\jre\lib\security\slkeystore.>
```

```
<Notice> <Server> <BEA-002613> <Channel "DefaultSecure" is now listening on 192.168.21.72:9443 for
protocols iiops, t3s, ldaps, https.>
```

Your Service Link service is now SSL-enabled.

Step 11 Skip this step if you have already created the file "slsigner.cer" that contains the signer certificate for the *servicelink* certificate. Otherwise, you can perform the following procedure to export the signer certificate. There are several methods to export the signer certificate; the following procedure is just one way to do it using the "keytool.exe" utility that comes with the Sun JDK 6 installation.

a) Execute the following commands on a Command Prompt window:

Example:

```
cd <JAVA_HOME>\jre\lib\security
<JAVA_HOME>\bin\keytool -export -rfc -file slsigner.cer -alias servicelink -keystore slkeystore.jks
 -storepass slpassword
```

b) To verify that file "slsigner.cer" is good, execute:

Example:

```
<JAVA_HOME>\bin\keytool –printcert –file slsigner.cer
```

Step 12 If you decide to disable the nonsecure port for the Service Link service, send the file "slsigner.cer" to the system administrator who manages the external system which communicates with the Service Link service. Two things will need to be configured for that external system:

a) The Service Link URL must be changed from http to **https** address with the secure port number. For example, previously, the Service Link URL may be:

Example:

```
http://<sl_servername>:9001/IntegrationServer/ishttplistener/ <agent_name>
```

It must be changed to:

Example:

```
https
://<sl_servername>:9443
/IntegrationServer/ishttplistener/<agent_name>
```

b) The signer certificate of the *servicelink* certificate (i.e. the contents of file "slsigner.cer") needs to be imported into the *Java Trusted Certificate Authority Keystore* of the external system, so that a trusted handshake can be established during the SSL connection with the Service Link service.

For a clustered WebLogic environment

Step 1 environment To disable the nonsecure port for the Service Link service, you must import the signer certificate into the *Java Trusted Certificate Authority Keystore* of the Service Catalog service.
This is because Service Link runs a separate WebLogic server that does not belong to the cluster. (Only Service Catalog and the Business Engine can be installed on the cluster.) Service Catalog acts as a "client" that connects to the Service Link service at runtime.

Step 2 Complete the following procedure to import the signer certificate into the *Java Trusted CA Keystore* for Service Catalog:

a) Log on to one of the nodes of the WebLogic cluster where Service Catalog application is running.

b) Locate the file "**cacerts**" in the directory "<*JAVA_HOME* >\jre\lib\security", where <*JAVA_HOME* > is the root directory of the Sun JDK 6 installation. This file is the Trusted CA Keystore that comes with the Sun JDK 6 installation. Make sure that <JAVA_HOME> is the correct Java directory used by your WebLogic application server. To verify this, look for the JAVA_HOME setting inside file "commEnv.cmd" (on UNIX/Linux, look for "commEnv.sh"), located under the "<WL_HOME>\common\bin" directory. For example:

Example:

```
set JAVA_HOME=C:\jdk170
```

c) Copy the file "slsigner.cert" to the "<*JAVA_HOME* >\jre\lib\security" directory.

d) Import the signer certificate into the "cacerts" keystore by executing the following commands on a Command Prompt window:

Example:

```
cd <JAVA_HOME>\jre\lib\security
<JAVA_HOME>\bin\keytool -import -trustcacerts -alias servicelink –noprompt -file slsigner.cer
-keystore cacerts -storepass changeit
```

In the command above, the password for the "cacerts" keystore file is still the default value of "changeit". Replace it with the correct value if the password for "cacerts" has been changed in your environment.

e) Copy file "cacerts" that you just updated in the last step to the "<*JAVA_HOME* >\jre\lib\security" directory on every node in the WebLogic cluster where Service Catalog is deployed. For example, if your WebLogic cluster contains three nodes, and each node is a separate machine, then copy the file "cacerts" from this machine to the other two machines.

f) Modify file "**newscale.properties**" under the directory "<*BEA_HOME* >\ user_projects\ domains\<*domain_name* >\servers\<*servername* >\stage\ RequestCenter\config" as follows:
Search for the following parameter:

Example:

```
isee.base.url=http://<hostname>:9001
```

and change it to:

Example:

```
isee.base.url=https
://<hostname>:9443
```

g) Repeat Step (f) for every node in the WebLogic cluster where Service Catalog is deployed.

h) Restart the WebLogic cluster for Service Catalog.

To avoid Step 1 entirely, you may decide to turn on both the nonsecure and secure ports for the Service Link service. This way the Service Catalog application can still connect to Service Link using the nonsecure URL (http://<*hostname* >:9001), however, ou may want to consider taking some measures (such as a firewall system) to block access to the nonsecure port from all external systems.

Configuring SSL for Service Link Outbound Documents

When a Service Link agent uses the HTTP/WS adapter to send an outbound message to an external system, it acts as a client that posts http requests or web services request to the external web server. If the external web server is SSL-enabled, Service Link may require some configuration in order to establish a secure connection with that web server.

- The Outbound URL of the Service Link agent must point to the https address with the secured port number of the external web server.
- To establish a trusted handshake via SSL, the client (that is, the Service Link service) must have a valid signer certificate (the public key certificate) that can validate the digital certificate of the external web server. If the certificate of the external web server is not signed by a well-known Certificate Authority (CA) such as VeriSign, then most likely during the SSL handshake, Service Link will not be able to validate the external web server certificate, and the connection will fail. If this is the case, the signer certificate must be imported into the *Trusted Certificate Authority Keystore* used by the Service Link service.

Note If Service Link is connecting to multiple SSL-enabled web servers, it may be necessary to import multiple signer certificates, one for each external web server.Service Link, as a client, does not support Client Certificate Authentication during SSL handshake.

The following sections describe the configuration procedure in detail.

- Specifying the Outbound URL for SSL, on page 44
- Importing the Signer Certificate to a Trusted CA Keystore, on page 45
- Configuring JBoss 7.1.1, on page 45
- Configuring WebLogic 10.3.6 (11g), on page 46

Specifying the Outbound URL for SSL

Step 1 Log on to Cisco Prime Service Catalog as a user who can access Service Link, navigate to the Service Link module and click the **Manage Integrations** tab.

Step 2 Choose the agent that you want to configure, open the Outbound Properties page of the agent.

Step 3 In the **HttpOutboundAdapter.RoutingURL** field, enter the https address with the secured port number, for example, https://192.168.21.202:8444/HTTPSimulator/.

Step 4 Set the value for the **HttpOutboundAdapter.AcceptUntrustedURL** field to **false** to ensure a secure connection.

Step 5 Click **Save**, open the Control Agents tab, and restart the agent.

Importing the Signer Certificate to a Trusted CA Keystore

Before following the application server-specific instructions, you must complete the following step:

Note If the signer of the external web server certificate is a well-known Certificate Authority like VeriSign or Thawte, then most likely, you can skip this step since Sun JDK already recognizes CA signers.

- Obtain the signer certificate of the external web server in a file. To do this, you can contact the system administrator who manages the external web server, and ask him/her to export the signer certificate (the public key) of the digital certificate used to secure that web server. The signer certificate must be exported in the "**Base64-encoded ASCII**" format. The following is an example of what a Base64-encoded signer certificate looks like:

```
-----BEGIN CERTIFICATE-----
MIICPDCCAaUCBE17w1cwDQYJKoZIhvcNAQEEBQAwZTELMAkGA1UEBhMCVVMxCzAJBgNVBAgTAkNB
MRIwEAYDVQQHEwlTYW4gTWF0ZW8xETAPBgNVBAoTCG5ld1NjYWxlMQswCQYDVQQLEwJRQTEVMBMG
A1UEAxMMS2hhbmcgTmd1eWVuMB4XDTExMDMxMjE5MDI0N1oXDTIxMDMwOTE5MDI0N1owZTELMAkG
A1UEBhMCVVMxCzAJBgNVBAgTAkNBMRIwEAYDVQQHEwlTYW4gTWF0ZW8xETAPBgNVBAoTCG5ld1Nj
YWxlMQswCQYDVQQLEwJRQTEVMBMGA1UEAxMMS2hhbmcgTmd1eWVuMIGfMA0GCSqGSIb3DQEBAQUA
A4GNADCBiQKBgQDhTxg2RwarD6Wn4iqYeOOk3ykfXzZiDArf/X63omXquTmN0Up+mg6oJmPAfqJA
l7k4+Dn7dfVtAc4h8qra7PBeBU48zrzRqZd6VAK07rz++CilQto64mHXYVomb5vWPGeKA41j9vlv
ENj/tE/6++IqbwnxAqeZtY3EvEM7dcCWDwIDAQABMA0GCSqGSIb3DQEBBAUAA4GBAAaqCnfEAovy
Uf2S+oAXYDo5N387a035APsz5iiUM5oiKR/KW3oRz/v0P0I/o3n312kDIJ01111pl6qpZRtPEsr1
b00Tu1cXfPmizEtz0ole606qDS+DzkS1+YYz2mLL2Zq40d1EPsMolyqyUmyq3GHaEnuhWemcv2aA
wGFgbQYd
-----END CERTIFICATE-----
```

The instructions for importing the signer certificate depend on the application server (Configuring JBoss 7.1.1, on page 45, Configuring WebLogic 10.3.6 (11g), on page 46, or Troubleshooting, on page 46) that Service Link is running on.

Configuring JBoss 7.1.1

Perform the following steps as the "administrator" user of the Service Link machine:

Step 1 Copy the signer certificate file (of the external system) to a temporary directory on the Service Link machine. For example, if the signer certificate file is called "extws.cer", then copy this file to "C:\temp\extws.cer" on the Service Link machine.

Step 2 On the Service Link machine, locate the file "**cacerts**" in the directory "<*JAVA_HOME* >\jre\lib\security", where <*JAVA_HOME* > is the root directory of the Sun JDK 6 installation. This file is the *Trusted CA Keystore* that comes with the Sun JDK 6 installation.

Step 3 Import the signer certificate into the "cacerts" keystore by executing the following commands in a Command Prompt window or a Console window:

Example:

```
cd <JAVA_HOME>\jre\lib\security
<JAVA_HOME>\bin\keytool -import -trustcacerts -alias extws –noprompt  -file C:\temp\extws.cer -keystore
 cacerts -storepass changeit
```

Note In the “keytool” command above, it is assumed that the password for the “cacerts” keystore file is still the default value of “changeit”. Replace this with the correct value for the password in your environment. For the –alias parameter, you can replace the value “extws” with an appropriate alias you plan to use for this signer certificate. If you import multiple signer certificates, make sure to assign a unique alias name to each signer certificate.

Step 4 Restart the Service Link service.

Configuring WebLogic 10.3.6 (11g)

Perform the following steps as the “root” user (if on UNIX/Linux) or the “administrator” user (if on Windows) of the WebLogic machine:

Step 1 Copy the signer certificate file (of the external system) to a temporary directory on the WebLogic machine where Service Link service is running. For example, if the signer certificate file is called “extws.cer”, then copy this file to “/tmp/extws.cer” on the Service Link machine.
In a clustered WebLogic environment, Service Link must be deployed in a WebLogic server that does not belong to the cluster. So, make sure you find the correct WebLogic server for Service Link.

Step 2 On the Service Link machine, locate file “**cacerts**” in the directory “<JAVA_HOME>/jre/lib/security”, where <*JAVA_HOME* > is the root directory of the Sun JDK 6 installation. This file is the *Trusted CA Keystore* that comes with the Sun JDK 6 installation.
Make sure that <JAVA_HOME> is the correct Java directory used by your WebLogic application server. To verify this, look for the JAVA_HOME setting inside file “commEnv.sh” (on Windows, look for “commEnv.cmd”), located under the “<WL_HOME>/common/bin” directory. For example: JAVA_HOME=“/opt/jdk1.6.0_23”.

Step 3 Import the signer certificate into the “cacerts” keystore by executing the following commands in a Command Prompt window:

Example:

```
cd <JAVA_HOME>/jre/lib/security
<JAVA_HOME>/bin/keytool -import -trustcacerts -alias extws –noprompt  -file /tmp/extws.cer -keystore
 cacerts -storepass changeit
```

Note In the “keytool” command above, it is assumed that the password for the “cacerts” keystore file is still the default value of “changeit”. Replace this with the correct value for the password in your environment. For the –alias parameter, you can replace the value “extws” with an appropriate alias you plan to use for this signer certificate. If you import multiple signer certificates, make sure to assign a unique alias name to each signer certificate.

Step 4 Restart the WebLogic server where Service Link is deployed.

Troubleshooting

This section provides information about how to limit outbound email and to control email generation. It also includes information about contacting Cisco with support questions and methods for keeping track of your system environment and error information.

Tracking and Troubleshooting Application Provisioning Process

After you order the application template, the orchestration component provides an option to track the template provisioning progress in **Comments and History**, under My Stuff.

If the (built-in) Cloud orchestration service is restarted when Prime Service Catalog is running, it reconnects to Prime Service Catalog, discovers AMQP exchanges, and resumes monitoring of AMQP messages.

Whereas, if Prime Service Catalog is restarted when Cloud orchestration service is running, the Cloud orchestration service reconnects to Prime Service Catalog when it resumes execution.

When an order (for the application template) is submitted, the Cloud orchestration engine (Heat Engine) status is checked before the engine starts provisioning the application template:

If the Cloud orchestration engine or the Cloud orchestration engine API service is down, the Cloud orchestration service cancels the requisition in Prime Service Catalog and logs in **Comments** for that requisition: `Heat Engine service is down. Details: <More information on the service status>.`

Note Cloud orchestration engine is not fault tolerant: If it goes down when an infrastructure template is being provisioned, the provisioning is halted and cannot be recovered when the engine is restarted at a later point.

Restarting Cloud Orchestration Engine and Orchestration services

Enable root access from the Shelladmin Menu, login as root (using the Shelladmin menu option), run the following commands, view log files and so on. (Alternatively, using the Display Service Status option, you can view the status for all services including the following services):

```
sudo service openstack-keystone restart
sudo service openstack-heat-api restart
sudo service openstack-heat-api-cfn restart
sudo service openstack-heat-engine restart
sudo service amqp-service restart
sudo service psc-orchestration restart
```

Log Files

You can examine orchestration service and heat engine logs under **Administration > Utilities > Logs and Properties**, and choose **Request Center - Log Files**.

- Orchestration logs are located in /var/log/cisco/psc/psc-orchestration.log
- Cloud Orchestration (Heat) engine logs are located in /var/log/heat/engine.log

Commonly Monitored Traces

The following traces are commonly monitored:

Database Interactions	com.newscale.bfw.udkernel.udsql.UdSqlBean com.newscale.bfw.udkernel.util.UdKernelUtil
LDAP Interactions	com.newscale.bfw.ldap.jldap.JLDAPApi
Clustering Issues	net.sf.ehcache.distribution.jgroups.JGroupsCacheManagerPeerProvider

Limiting Outbound Email

You may want to limit outbound email during service design testing or in nonproduction environments.

By limiting outbound email capabilities, you can limit or prevent the sending of email to actual performers or customers on whose behalf services are ordered.

Changing all email templates to have “fake addresses” in a development environment is not really an option. Firstly, it would be very time consuming. More important, much of the testing is invalidated when the template addresses are changed back—you would still need to ascertain that the correct people are receiving the appropriate emails.

If templates use only namespace variables and users in the nonproduction environment are refreshed via directory integration, you could change the LDAP mapping to give everyone the same email address or a similar fake address, for example:

User@<company>.com, or

reqcentertest@<company>.com

by using a mapping similar to:

```
expr:#cn#=(cannotmatch)?(neverthis):requestcenter@<company>.com
```

However, this approach also does not allow you to adequately test the accuracy of email delivery.

A more robust solution is to use a dedicated SMTP (email) server for the development instance and any other instances where emails should not be distributed outside the box. You can set up an SMTP server that routes ALL emails (whether fake or correct) to a standard mailbox (for example, rctestmailbox@company.com) for the development and test servers. This way, you do not have to change Service Catalog configuration in any way, and emails could be tested very easily. The project team just needs to be able to open that test mailbox.

This requires users to configure a separate test SMTP server that overrides the recipients to always forward to the test email box. Production would need to point to the production SMTP server, of course.

If you use any of these techniques, add the To/Cc addressees in the HTML body of the email templates surrounded by <!-- Comment --> tags so that testers may validate the namespace expression and other logic for these fields.

Controlling Email Generation

Service Catalog controls the outgoing email envelope and defaults to sending a single message to multiple recipients. The multiple-recipient messages are sent to the same SMTP server.

The alternative is to send single recipient emails as it has a minimal negative effect on CPU and network bandwidth usage. This is enabled via a setting in the newscale.properties file:

```
Email.One.Per.Recipient=true
```

Use this setting only to avoid SMTP server problems whereby the entire message is rejected if one recipient is invalid.

SMTP Connections are tried 10 times (by default) and are configured by the Email.ServerDownCount property. The connection retries to the SMTP host are paused for the configured time (in msec) specified in the Email.RescheduleOffset property.

In addition, issues such as configured mailbox exceeding the set limit, email bounces, or other delivery problems are retried based on the default setting of the Email.RetryCount property (currently, the default is 4).

Environment/Platform Overview

It is useful to document the systems in your environment by using a matrix like the one provided in the Sample Environment Matrix, on page 62.

Cisco publishes a support matrix detailing the software on which each version of Service Catalog is certified. The Cisco Technical Assistance Center (TAC) will always have the most current version of this matrix, adjusted for point releases and Service Packs.

Contacting Cisco Technical Assistance Center (TAC)

You can inform the Cisco Technical Assistance Center (TAC) before performing any system maintenance tasks that may affect:

- Server operating system patches/upgrades
- Database server patches/upgrades
- Service Catalog application server patches or upgrades – Validate the update is supported by Cisco first!
- LDAP Directory tree structure changes
- Single Sign-On system upgrades

Collecting Troubleshooting Information

This section describes about gathering troubleshooting information in various situations.

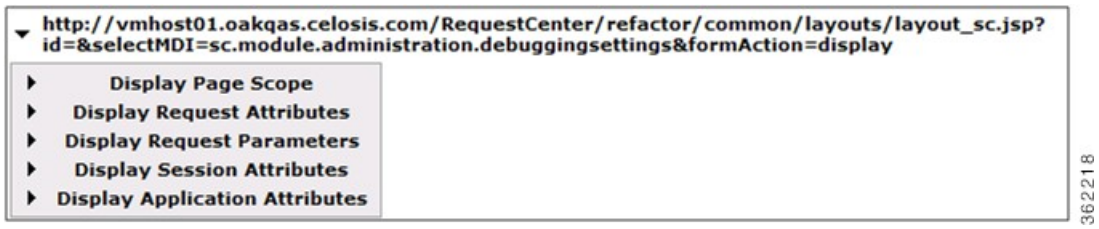
Site Debugging

If an “Our Apologies” exception occurs, you may turn on “Debug” via the Debugging option of Administration module Settings.

Figure 1: Debugging page

Debugging adds the URL of the current page at the bottom of the page. Clicking on the URL provides links to additional information which may be helpful to Cisco support personnel.

Figure 2: URL at the bottom



When you are finished, turn off the debugging, as it may confuse end-users. It also adversely affects performance.

The application log is a key troubleshooting mechanism. Checking this log for “Exception” (from the bottom up) often reveals the applicable error message.

In a clustered environment, it is often useful to browse the log files from all the machines in the cluster for the period in question.

Service Link Log Files

Logs for the Service Link server show the details of all Service Link transactions for that day. It is often useful to correlate that file to the Service Catalog server log when troubleshooting issues that have to do with the interaction between the Business Engine and Service Link.

Performance

Gather performance information from the log and native_stderr.log files.

Service Design and Platform Dependence

Problems that arise during service design may be related to incorrect service configuration. Problems that occur only in a production environment may be data-dependent or platform-dependent.

In some cases, the Cisco Technical Assistance Center (TAC) may ask for a dump of the database to be sent, where it can be installed in a testing lab that can closely emulate the environment where the error occurred. Customers should have logins and credentials that allow them to upload the database to the Cisco support site for investigation.

Contact the Cisco Technical Assistance Center (TAC):

- For Solutions
  - Get access to the documentation library
  - Learn about upgrades and patches
  - Learn answers to Common Issues
- About Cases
  - Log new cases
  - Check status of cases
  - Read/Update case investigation comments

- Attach logs/files

Enabling Adapter Log Files for ServiceLink Application

On WebLogic, the log files for the ServiceLink adapters are not enabled by default. By default, all logging for ServiceLink adapters are written to the server.log file for the WebLogic server.

This section describes the configuration steps to enable the adapter log files for ServiceLink. These configuration steps must be performed manually by the user after the ServiceLink WAR is deployed.

Note This section is not applicable for JBoss, since the Service Catalog Installer automatically configures the ServiceLink adapter log files at installation time.

For WebLogic 11g

Step 1 If the ServiceLink application is deployed and running on the WebLogic server, stop the WebLogic server. You cannot stop just the ServiceLink application; you must stop the entire WebLogic server.

If you have not deployed the ServiceLink application, then follow the steps up to the point where you have to extract "ISEE.war" into a ServiceLink directory. (Remember that you must deploy ServiceLink in an extracted WAR format.) Next, perform the steps described in this section in the extracted ServiceLink directory, before you begin the deployment. In other words, in Step 3 below, you navigate to the extracted ServiceLink directory, instead of the staging directory.

Step 2 Log in to the machine where ServiceLink WAR is deployed.

Step 3 Navigate to the directory "<BEA_HOME>\user_projects\domains\<domain_name>\servers\<server_name>\stage\ServiceLink\WEB-INF\classes\config".

Step 4 Use a text editor to modify file "newscalelog.properties" as follows:

1. Make sure that the line "logger.class.name=com.newscale.bfw.logging.LogUtilCommonsImpl" is not commented out.
2. Remove the comment sign in front of "logger.directory=", then enter the correct log directory for the WebLogic server where ServiceLink application is deployed. This should be the directory where the "server.log" for the WebLogic server is located. For example,

On UNIX or Linux:

logger.directory=/opt/bea/user_projects/domains/mydomain/servers/server1/logs

On Windows:

logger.directory=C:/bea/user_projects/domains/mydomain/servers/server1/logs

Note On Windows, use the slash (/), instead of the backslash (\) as the directory delimiting character.

Step 5 Start the WebLogic server.

In the same directory for "server1.log", you can see a new log file called "isee.log", and several additional log files – one for each ServiceLink adapter.

Errors

This section provides information regarding critical error conditions. The information is presented according to individual error messages, and includes the following information for each condition:

- Error Condition
- Error Message
- Probable Cause
- Location of Error Log
- Recommended resolution

See also, Managing Logs, on page 22.

Error Log Locations

Error logs for Service Catalog and its related components are in the following locations:

Table 13: Error log path

Component	Error Log Location
Application Server	
WebLogic	<BEA_HOME>/user_projects/domains/<domain>/servers/<server>/logs/<server>.log
JBoss	<JBOSS_HOME>/ServiceCatalogServer/log, <JBOSS_HOME>/ServiceLinkServer/log

If you have configured the support utilities in Administration module to enable GUI access to the application log files, you can also view and download the above log files from there.

Error Conditions and Error Codes

The following error conditions are presented according to the error condition or its related error message.

Some error conditions cause the same system behavior although the error itself may stem from one of several different error conditions within the system. For example, if you cannot connect to the LDAP server, several error conditions below may apply. It is important to match the error message to the error you are experiencing.

All errors are written to the Service Catalog server log file, whose behavior and location are described earlier.

Failure to perform Asynchronous Submit/Authorization

Table 14: Submit/Authorization errors

Category	Description
Error Condition	Service Catalog is not able to instantiate a task plan asynchronously, after the request submission or the last authorization/review in the service.
Error Message	Requisition *xxx* [Task "*<name of task here >*"]: We're sorry but his approval/review cannot be completed at this time because the Service Catalog queue that processes these tasks is temporarily unavailable. Please try again later or contact your Service Catalog system administrator.
Resolution	Verify that the JMS queue which serves the asynchronous submit/last authorization process is available for receiving messages.

Application Server Loses Connection to the Database

Table 15: Connection loss errors

Category	Description
Error Condition	Application server lost the connection to the database.
Error Message	ERROR [com.celosis.logger.FatalerrorChannel] (8000)SQLException in getConnection:Could not create connection; - nested throwable: (java.sql.SQLException: [newScale][SQLServer JDBC Driver]Error establishing socket. Connection refused: connect); - nested throwable: (org.jboss.resource.JBossResourceException: Could not create connection; - nested throwable: (java.sql.SQLException: [newScale][SQLServer JDBC Driver]Error establishing socket. Connection refused: connect)) Code: 0 State: null.
Resolution	Check the RequestCenter database. If the RequestCenter database is not running, start it. Once the database is up, the application server will automatically connect to it.

Failure to Connect to the LDAP Server – Incorrect Port

Table 16:

Category	Description
Error Code	LDAPException 91.
Error Condition	Cannot connect to the LDAP server. Most likely the LDAP server is down or you have an incorrect port number.
Error Message	ERROR [com.newscale.bfw.ldap.jldap.JLDAPNonSSLConnection] LDAPException in NON-SSL Connection: LDAPException: Unable to connect to server <hostname>:<port> (91) Connect Error java.net.ConnectException: Connection refused: connect
Resolution	Check to see if the LDAP server is running. If not, start the LDAP server. Check the LDAP System Connection Parameters on **Administration > Directories**. Verify that the Connection Port value is correct. You do not need to restart the Service Catalog application.

Failure to Connect to the LDAP Server – Incorrect Hostname

Table 17:

Category	Description
Error Code	LDAPException 91
Error Condition	Cannot connect to the LDAP server. Most likely incorrect hostname.
Error Message	ERROR [com.newscale.bfw.ldap.jldap.JLDAPNonSSLConnection] LDAPException in NON-SSL Connection: LDAPException: Unable to connect to server <hostname>:<port> (91) Connect Error java.net.UnknownHostException: <hostname>

Category	Description
Resolution	Check the LDAP System Connection Parameters on **Administration > Directories**. Verify that the LDAP Host value is correct.

Failure to Connect to the LDAP Server – LDAPException 32

Table 18:

Category	Description
Error Code	LDAPException 32
Error Condition	Cannot connect to the LDAP server. Most likely incorrect authenticated user id.
Error Message	ERROR [com.newscale.bfw.ldap.jldap.JLDAPSimpleAuth] LDAPException in Simple Auth: LDAPException: No Such Object (32) No Such Object LDAPException: Matched DN:
Resolution	Check the LDAP System Authentication Parameters on **Administration > Directories**. Verify that the BindDN value is correct.

Failure to Connect to the LDAP Server – LDAPException 49

Table 19:

Category	Description
Error Code	LDAPException 49
Error Condition	Cannot connect to the LDAP server. Most likely incorrect authenticated password.
Error Message	ERROR [com.newscale.bfw.ldap.jldap.JLDAPSimpleAuth] LDAPException in Simple Auth: LDAPException: Invalid Credentials (49) Invalid Credentials

Category	Description
Resolution	Check the LDAP System Authentication Parameters on **Administration > Directories**. The Password field is encrypted and thus you can not verify its existing value. Just enter a correct value for the Password, and click **Update**.

Failure to Connect to the LDAP Server

Table 20:

Category	Description
Error Condition	Cannot connect to the LDAP server.
Error Message	FATAL [LDAPBase] LDAP instance cannot be created netscape.ldap.LDAPException: no host for connection (89)
Resolution	Check to see if the LDAP server is running. If not, start the LDAP server. You do not need to restart the Service Catalog application server.

Failure to Connect to the LDAP Server

Table 21:

Category	Description
Error Condition	Cannot connect to the LDAP server.
Error Message	ERROR [com.newscale.bfw.ldap.LDAPQuery] LDAP netscape.ldap.LDAPException: failed to connect to server ldap://<hostname>:<port> (91)
Resolution	Check to see if the LDAP server is running. If not, start the LDAP server. You do not need to restart the Service Catalog application server.

Failure to Authenticate with the LDAP Server

Table 22:

Category	Description
Error Condition	Fail to authenticate with the LDAP server.
Error Message	ERROR [com.newscale.comps.user.dao.LDAPUserDataSource] Single Person search failure, exception thrown: null com.newscale.bfw.dataaccess.DataAccessException
Resolution	Check the Data Source Configuration on the **Administration > Directories** page. Verify the following parameters and correct if necessary: • BindDN • Password • User BaseDN You do not need to restart the Service Catalog application server.

Attribute Name is Mapped Incorrectly

Table 23:

Category	Description
Error Condition	One of the required attributes is incorrectly mapped. Thus the person cannot be found in the LDAP server.
Error Message	ERROR [com.newscale.bfw.ldap.LDAPQuery] LDAP java.lang.RuntimeException: Required LDAP attribute <attribute_name> is missing from the LDAP system.
Resolution	Correct the attribute name in the Directory Data Mapping. You do not need to restart the Service Catalog application server.

User Base DN in LDAP Server is Missing

Table 24:

Category	Description
Error Code	LDAPException 32
Error Condition	Cannot find the User Base DN in LDAP server.
Error Message	ERROR [com.newscale.bfw.ldap.jldap.JLDAPApi] Referral Exception during Result Set iteration: LDAPException: No Such Object (32) No Such Object
Resolution	Check the LDAP System Authentication Parameters on the **Administration > Directories** page. Verify that the LDAP User BaseDN value is correct.

Failure to Connect to the LDAP Server in SSL Mode

Table 25:

Category	Description
Error Condition	(For SSL connection only) Cannot connect to the LDAP Server in SSL mode, because the SSL certificate keystore has not been created.
Error Message	DEBUG [com.newscale.bfw.ldap.util.LDAPConfUtil] The LDAP configuration file “config/<LDAP_System>_TrustCertDB.keystore” does not exist.
Resolution	Add the appropriate server certificate for the LDAP System on the **Administration > Directories** page.

Failure to Connect to the LDAP Server in SSL Mode

Table 26:

Category	Description
Error Condition	(For SSL connection only) Cannot connect to the LDAP Server in SSL mode, because the server certificate in the keystore is NOT correct.

Category	Description
Error Message	ERROR [com.newscale.bfw.ldap.jldap.JLDAPSimpleAuth] LDAPException in Simple Auth: LDAPException: I/O Exception on host <hostname>, port <port number> (91) Connect Error javax.net.ssl.SSLException: Connection has been shutdown: javax.net.ssl.SSLHandshakeException: sun.security.validator.ValidatorException: No trusted certificate found
Resolution	The certificate keystore may already exist, but does not contain the correct certificate used with this LDAP Server. Obtain the correct certificate used for the LDAP server, and add it for the same LDAP System on the **Administration > Site Configuration** page.

"Common OU for new users" Configuration Value is Missing

Table 27:

Category	Description
Error Condition	The "Common OU for new users" configuration value is either missing or does not exist in RequestCenter database.
Error Message	ERROR [com.newscale.comps.user.dao.LDAPUserDataSource] Error getting Person from Ldap java.lang.NullPointerException at com.newscale.comps.user.dao.LDAPUserDataSource. transferOrgUnitVOToBO(LDAPUserDataSource.java:676)
Resolution	Check the LDAP System Lookup Configuration on the **Administration > Site Configuration** page. Choose a correct value for the "Common OU for new users" field.

User Cannot be Found in the LDAP Server

Table 28:

Category	Description
Error Condition	The <attribute_name> is incorrectly mapped. Thus, the person cannot be found in the LDAP server.
Error Message	WARN [com.newscale.bfw.ldap.jldap.JLDAPApi] Required LDAP attribute <attribute_name> is missing from the LDAP system, for DN : ...
Resolution	Correct the attribute name on the Directory Mapping page, for the appropriate LDAP System.

Failure to Connect to a Referral LDAP System

Table 29:

Category	Description
Error Condition	Cannot connect to one of the Referral LDAP Systems. (The config flag SkipErrorOnLDAPSystem=true; thus, Service Catalog system ignores this error.)
Error Message	WARN [com.newscale.bfw.ldap.jldap.JLDAPApi] Referral Exception during Result Set iteration: LDAPReferralException: Search result reference received, and referral following is off (10)
Resolution	Check to see if the Referral LDAP server is running. Verify the Authentication and Connection for the Referral LDAP System.

Failure to Connect to the External Data Dictionary Database

Table 30:

Category	Description
Error Condition	Cannot connect to the External Data Dictionary Database.

Category	Description
Error Message	ERROR [STDERR] SQLException while attempting to connect: java.sql.SQLException: [Macromedia][SQLServer JDBC Driver]Error establishing socket. Connection refused: connect.
Resolution	Check the External Data Dictionary database. If the External Data Dictionary database is not running, start it. You do not need to restart the Service Catalog application.

Lost Connection to the Database

Table 31:

Category	Description
Error Condition	Lost the connection to the database.
Error Message	ERROR [com.celosis.logger.FatalerrorChannel] (8000)SQLException in getConnection: Could not create connection; - nested throwable: (java.sql.SQLException: [newScale][SQLServer JDBC Driver]Error establishing socket. Connection refused: connect)
Resolution	Check the database. If the database is not running, start it. Once the database is up, the application server will automatically connect to it.

Failure to Connect to the External Data Dictionary Database

Table 32:

Category	Description
Error Condition	Cannot connect to the External Data Dictionary Database.
Error Message	ERROR [com.newscale.bfw.udkernel.udsql.UdSqlBean] Message: [newScale][SQLServer JDBC Driver]Connection reset by peer: socket write error.

Category	Description
Resolution	Check the External Data Dictionary database. If the External Data Dictionary database is not running, start it. You do not need to restart the Service Catalog application.

Sample Environment Matrix

It is a standard practice of the Universal Development Methodology (UDM) to complete a column in this matrix for each site in an implementation, as the site comes online. Cisco Advanced Services deliverables typically include a soft copy of this matrix, which administrators should keep up to date.

Table 33: Client Service Catalog Configuration

Category	Site Name/Usage (for example, Dev)
WebServer	
Front Door Cisco Prime Service Catalog URL	https://scdev/RequestCenter/
Admin Cisco Prime Service Catalog URL	https://scdevadmin/RequestCenter/
Host1:Port	
Shared Environment?	
Hardware	
Available Disk	
Operating System	
OS Login/Password	
WebServer Type/Version	
AppServer	
Host1	
Shared Environment?	
Hardware	
Available Disk	

Category	Site Name/Usage (for example, Dev)
Operating System	
Support Login/Pass	rcsupport/rc
Installer Login/Pass	requestcenter/rc
RC Path	/apps/rc
RC.ear Path	/apps/rc/RC.ear
ISEE.war Path	/apps/rc/ISEE.war
Log Path	/logs/rc
Queue Connection Factory	RCQueueConnectionFactory
BE Requisitions Queue	BEEERequisitionsQueue
BE Authorizations Queue	BEEEAuthorizationsQueue
BE Inbound Queue	BEEEInboundQueue
JDK	
JDK Path	/usr/local/java
App Container	
Type / Version	
AppHost1 RC/SL JNDI Ports	
Mail	
SMTP Server	smtpserver.domain.com
Administrator Email Address	
From Email Address	ServicePortalDev@mailserver.company.com
WebLogic	
Console URL	
User/Password	
Node Name(s)	

Category	Site Name/Usage (for example, Dev)
Application Server	RequestCenter
Virtual host	requestcenter_host
Service Catalog	
Components Installed	All
Multicast IPs	225.2.2.2
Build Installed	11.2.1.0151
Admin Login/Password	
Customizations	
Patches/Hotfixes applied	
Other customizations	
Database	
Host1:Port	
Shared Environment?	
Hardware	
Available Disk	
Operating System	
OS Login/Password	
DB Type/Version	
DB SID/Database	RQSTDEV
Tablespace	RequestCenter (?GB)
Redo logs	
DB SA User/password	sa/pwd
DB RC Schema/Password	RCUser/rc

Category	Site Name/Usage (for example, Dev)
DB App User/Password	
Advanced Reporting	
Cognos Host:Port	
Cognos Hardware	
Available Disk	
Cognos OS	Windows 2008
Windows Login/Pass	rcuser/c1$c0
Admin Login/Pass	admin/admin1234
Service Account	
Paths	
Gateway Type	
Web Protocol	
Data Mart & Content Store	
JNDI Name	java:/DATAMARTDS
DB Type/Version	
DB Server:Port	
DB SID/Name	RCDMDEV
Data Mart User/Password	DMUser/dm
ContentStore SID/Name	RCCSDEV
ContentStore User/Password	CSUser/cs
Tablespace	RCDataMart (500M)
Advanced Reporting Options	
Dictionary tables	150

Category	Site Name/Usage (for example, Dev)
Service tables	50
Dictionary table pattern	DM_FDR_DICTIONARYTABLE_
Service table pattern	DM_FDR_SERVICETABLE_
Field pattern	FIELD
Dictionary Text type fields	40
Dictionary Numeric type fields	10
Dictionary Date type fields	10
Service Text type fields	80
Service Numeric type fields	20
Service Date type fields	20
Text field max size	200
Refresh WDDX for any update	Yes/No
Service Link	
Host	localhost
Queue Host:Port	localhost:5099
Base URL	http://subdomain.domain.com:80
Queue Connection Factory	RCQueueConnectionFactory
Outbound Queue	SLOutboundQueue
Inbound Queue	SLInboundQueue
JMS Queue User/Password	guest/guest
JMS File Store (WLS-only)	ServiceLinkFileStore
JMS File Store (WLS-only)	
JMS Server	RCServer
LDAP	

Category	Site Name/Usage (for example, Dev)
Server Type	
LDAP Authentication	Simple
SASL Mechanism	—
BindDN	
BindDN Password	
Connection Mechanism	Non-SSL
SSL Type	—
LDAP Host	
Connection Port	389
Secure Port	—
LDAP User BaseDN	
Optional LDAP filter	