

Configuring Replication in Oracle Service Registry

Note: Source information for this document was provided by Systinet Corporation.

Replication Overview	1
Inter-Node Replication APIs Not Supported	1
Replication versus Federation	1
How Replication Works.....	1
Replication Basics.....	1
Mechanics of Replication	2
Configuring Replication.....	2
Master Registry Configuration	2
Slave Configuration	3

Replication Overview

Oracle Service Registry supports Selective One-Way Replication of new or changed businessEntity records from one or more Master Registry instances to a specified Slave Registry instance. This process can be used to support a variety of use cases where transfer of data from one Registry installation to another is desired.

INTER-NODE REPLICATION APIS NOT SUPPORTED

Note that this is NOT an implementation of the UDDI v3 Inter-Node Replication APIs as defined in the UDDI specification, which is not supported by Oracle Service Registry. The Inter-Node replication is intended to solve a different problem.

Instead, the replication mechanism in Oracle Service Registry is based on a UDDI subscription-based “pull” model.

REPLICATION VERSUS FEDERATION

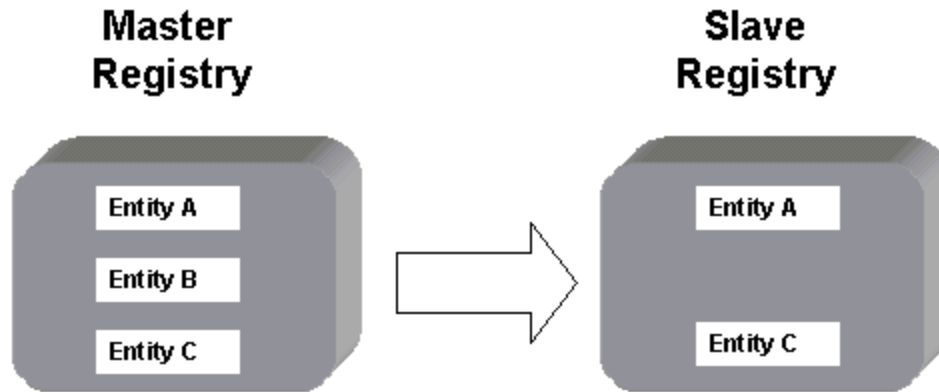
Note that the solution does not provide Federation capabilities, such as for a federated search where one registry propagates queries to other registries and consolidates the results.

How Replication Works

REPLICATION BASICS

Replications are simply a set of subscriptions created on the Master Registry that are then invoked from the Slave Registry.

Registry replication is designed to propagate a set of information from one or more Master Registry instances to a Slave Registry instance, as depicted in the figure below.



The registry administrator creates a Subscription on the Master Registry, which determines the set of information that is propagated from the Master into the Slave. The Replication is then in the Slave Registry, and determines how frequently the replication is performed.

By default, all new or updated businessEntities are copied to the Slave Registry. However it is possible to limit the data that is replicated using subscription filters configured on the Slave Registry. Note in the image above that not all entities that exist on the Master have been replicated to the Slave.

In addition, the Slave can have its own data which does not exist in the Master.

MECHANICS OF REPLICATION

At the time of execution, the Slave Registry performs a `get_subscriptionResults` call on the Master Registry based on the subscription key and time range (interval) specified in the replication.

For the purposes of this introductory section, keep in mind that the subscription on the Master **must** be a `find_business` subscription.

When the Slave runs the subscription, it gets back a set of keys corresponding to new or changed businessEntities. A businessEntity is considered to be changed if:

- Any of its businessServices or bindingTemplates have been modified, or
- A new businessService or bindingTemplate was created within it.

The Slave registry will then retrieve the complete businessEntity, and store it in the Slave Registry. If an instance of the businessEntity already exists on the Slave, it will be overwritten with the replicated data. As such, the Master is the “source of truth”.

Configuring Replication

MASTER REGISTRY CONFIGURATION

First you will create a Find Business subscription on the Master Registry. Make note of

the Subscription key, for use when configuring the replication (below).

1. Click the **Publish** link, then the **Subscriptions** link.
2. Click the **Add Subscription** button.
3. Click the **Change Filter** button on the Add Subscription page.
4. Select **Find Business Query** from the pull down menu.
5. Click **Select Filter**, then **Save Filter**.
6. Click the **Save** button.
7. Copy the subscription key, which you will use for configuring the replication.

SLAVE CONFIGURATION

Once the subscription has been created on the Master, you will configure the replication on the Slave Registry.

1. Log in to Registry Control as the Registry administrator on the Slave Registry.
2. Click the **Manage** tab.
3. Click the **Replications Management** button.
4. Click the **Add Replication** button.
5. Select the **Master** tab on the right.
6. Provide a login name/password for the user on the Master Registry that the Slave will authenticate as.

Ideally this should be the Master Registry Administrator account, as the user supplied must have the proper authorization to run the subscription and retrieve the results from the Master.

7. Enter the appropriate URLs for the appropriate APIs. Note that if the Master is a Publication Registry, you should supply the non-secure (non-HTTPS) URLs.

The syntax for each is:

```
<ohsHost>:<ohsPort>/<registryContext>/uddi/<apiEndpoint>
```

For example:

```
http://sta565:7778/registry/uddi/inquiry  
http://sta565:7778/registry/uddi/subscription  
http://sta565:7778/registry/uddi/security
```

8. Paste in the subscription key you copied for the subscription created on the Master Registry.

9. Now select the **Slave** tab on the right.
10. Enter a descriptive name for this replication
11. Enter the credentials for the Slave Registry Admin account. The replication must run under this account.
12. Configure the frequency at which the replication will run.
13. Finally, click the **Permissions** tab to optionally define ACL permissions to set on the replicated entities.

It is recommended that you grant read-only permissions to all users for replicated entities. This will avoid issues with overwritten modifications in the target registry.

INSERT SCREEN SHOT

Note that the Slave Registry Administrator owns all serviceEntities replicated to the Slave. The Administrator is therefore able to transfer ownership of a replicated service to a user on the Slave Registry using the Custody Transfer feature. This means that the accounts on the Master and Slave Registries do not have to be synchronized; instead, service ownership can simply be transferred to an existing user on the Slave.

Custom taxonomies are replicated with serviceEntities. However they are not exposable via the Slave Registry user interface, and taxonomy values are not available.

The best practice is to explicitly export taxonomies from the Master, then import the generated XML file containing the taxonomies into the Slave. You can then configure the Slave's Business Service Control user interface to expose these taxonomies in the Search and Publication pages.