

Creating and Configuring Replicated Vaults

To aid the administrators in deployment, each of the following topics is laid out in the order they should be considered to set-up Aras Innovator vault replication.

Obtain a Feature License

Vault replication requires a Feature License in order to run. Vault replication is enabled with the Activation key provided as a Subscription benefit.

If you do not have a current feature license, refer to *Feature License in Aras Innovator 12 – Installation Guide* for instruction on obtaining one.

Install Aras Agent Service

Vault Replication requires the Aras Agent Service in order to run. For instructions on installing the Aras Agent Service, see Section 5.12 of the *Installation Guide*.

Configure Alternate Vault Installations

The first step in setting up vault replication is to create the alternate vaults, if they do not already exist. Refer to **Configuring Alternate Vaults Installations** section in the *Aras Innovator - Installation Guide* for instructions to set up each vault.

Replication Process Configuration

Aras Innovator must periodically check for replication transactions in the queue that are due to run. There are two ways to configure the server running Aras Innovator to execute replication processing – either (a) within the Aras Innovator Server as a thread or (b) outside of Aras Innovator as an Aras Innovator service/scheduler or a program configured with standard Windows scheduler.

Each option has pros and cons. The advantage of the option (a) is that there is no need to configure an additional external service in order to initiate the processing of the replication queue. The disadvantage of the option (a) is where the Aras Innovator server is installed in a cluster environment



the queue processing thread on each server of the cluster might compete with each other which might cause certain problems.

Besides scheduled replication transactions, some transactions may be created with no scheduled execution time, and are 'Manual' replication transactions. These are run through a special method call described later in this section. Also, see *Programming Notes – Manual Replication Transactions* for more information on programming.

Within Aras Innovator Server – Replication Configuration File

If you choose to have the Aras Innovator server run a special thread that handles the replication queue, the thread depends on a set of parameters (e.g. 'queue processing interval, etc.). When the Aras Innovator server starts it reads a replication configuration file (called replication.config) that defines replication parameters.

The replication.config file is located in, and must reside in the Aras Agent Service. It is initialized to be disabled, and must be enabled, and populated with appropriate configuration settings to start replication with this option.

Because one Aras Innovator server can support multiple databases the configuration file might contain several nodes (one per database) with replication parameters. If a database supported by the Aras Innovator server does not have a corresponding node in the replication.config file, then files referenced from the database are never replicated. The replication parameters include:

- Replication interval is a time interval in seconds between consecutive invocations of the replication processing thread that goes through replication transactions and executes transactions that must be executed. The parameter is optional with the default value 60 sec.
- User name is the user, who is used by the processing thread for making requests to vault servers. The user is required purely for authentication purposes as any request to a vault server must have a valid user credentials set in the request header. All claiming/update/etc. operations are done by the processing thread on behalf of the user but with granting the user administrative privileges using GrantIdentity(...). Based on the above, it's recommended to create a special vault replication user in Aras Innovator with minimal privileges (i.e. who belongs only to World identity) and specify this user in the replication configuration file. This is a parameter with the default value: user=vadmin. If you use the default, make sure you have a user with a vadmin username, otherwise, the routine does not run.



- Max (Transactions per) Batch and Maximum (Transactions) Pending are the maximum amount of replication transactions that could be executed in a single ProcessReplicationQueue request (max_batch) and maximum amount of pending transactions allowed at any time (max_pending). If not specified, the server uses the default values for those parameters which are max_batch=25 and max_pending=50.

Important

The default values were picked based on experiments performed using Windows 2012 Server on a machine with four processors and 4 GB of memory. If the machine running the Aras Innovator server has more processors and more memory and none of vault servers are run by the same IIS worker process as the Aras Innovator server these numbers could be increased.

Based on testing, it is recommended to pick these numbers roughly as: max_batch = number of processors on the server * 10 max_pending = max_batch*2 If these numbers are too high, the IIS may hang up. The replication.config file is considered to be active only if it contains the attribute status that has value enabled. In all other cases the configuration file is considered as inactive and the replication queue processing thread is not started. The format of the replication.config file is the following (optional parameters shown in []), and where {db name} is the name of the Aras Innovator database:

```
<replication status="enabled">
<rnode db="{db name}" [user="..."] [interval=".."] [max_batch=".."] [max_pending=".."] />
...
</replication>
```

Given that format, and example replication.config file could look like this:

```
<replication>
<rnode db="Innovator" user="vadmin" interval="90" max_batch="30" max_pending="60" />
</replication>
```

Important

After making adjustments to the replication.config file, the Aras Innovator Agent should be restarted via **Control Panel --> Administrative Tools --> Services**.

Processing Queue Controlled Outside Aras Innovator Server

The queue processing can be initiated by sending to the Aras Innovator server a request with SOAP action ProcessReplicationQueue. The following AML may be passed to the SOAP action: <Item type='ReplicationTxn' [max_batch='...'] [max_pending='...'] /> where max_batch is the maximum



amount of replication transactions that could be executed in a single ProcessReplicationQueue request and max_pending is the maximum amount of pending transactions allowed at any time. If not specified, the server uses the default values for those parameters which are max_batch=25 and max_pending=50. The returned result XML has the following format: `<Result> <Item type= "ReplicationTxn " [processed= " {n} " failed= " {n} " need_processing= " {n} " locked_by_others= " {n} " /> ] </Result>` (inside the standard `<SOAP-Envelope> <SOAP-Body>` tags), where attributes have the following meaning:

- **processed** - integer number of transactions successfully processed.
- **failed** - tells how many of transactions among processed transactions failed and were assigned status Failed. A typical situation why a replication transaction processing can fail at this stage is, for instance, when the vault server to which the replication transaction was sent is not accessible.
- **need_processing** - integer that tells how many transaction in the queue are ready to be processed but were not processed in this processing cycle (e.g. because number of currently pending transactions were equal to max_pending).
- **locked_by_others** - integer that tells how many of transactions that need processing (see need_processing above) cannot be executed because the file is claimed not by the user on which behalf the replication queue processing is performed but by someone else. If claimed_by_others equals need_processing it usually means that the processing loop should be stopped because no more transactions could be executed unless user(s) who claimed those file unclaim them.

Important

Do not call this server method repeatedly on the server only within Aras Innovator, since Aras Innovator server calls get combined into one transaction which is committed only upon exit.

Important

In Aras Innovator, the nash utility now has an additional SOAP action 'ProcessReplicationQueue' which can be executed for development testing purposes.

Processing Manual Replication Transactions Programmatically

Replication rules can be configured to make some replication transactions not have a scheduled execution time. These transactions have a Replication Mode set to Manual. If any manual



transactions exist, they are held until executed programmatically through a call to a special new server method in the form - `Aras.Server.Replication.Queue.Process(XmlDocument inDom,XmlDocument outDom)`. `inDom` has the XML format: `<Item type="ReplicationTxn" [max_batch="..."] [max_pending="..."] />` , as shown in section [2.4.2](#) . The returned `outDom` XML has the following format: `<Result> <Item type="ReplicationTxn" [processed="{n}" failed="{n}" need_processing="{n}" claimed_by_others="{n}"/></Result>` (inside the standard `<SOAP-Envelope> <SOAP-Body>` tags), Definitions of each are shown in section [configaltvault](#) .

Important

Do not call this server method repeatedly on the server only within Aras Innovator, since the Aras Innovator server calls get combined into one transaction which is committed only upon exit.

See *Programming Notes – Manual Replication Transactions* for more information on programming.

