



CASE MANAGER 5.2 - INSTANCE DIRECTORY

TECHNICAL MANUAL

Table of Contents

OVERVIEW AND PURPOSE	4
DEFINITIONS	4
Instance	4
Default Instance	4
Instance Directory	4
Instance Directory Windows Service	4
Local Instance Directory	4
Instance Directory Utility	4
Policy	5
Update	5
Selection	5
Fail-over Policy	5
HOW DOES IT WORK?	5
Overview	5
Figure 1: Database connection initialisation overview	5
Start-up	5
Local Instance Directory Update	5
Instance Selection	5
Database initialisation	6
Fail-over	6
Application	6
Instance Directory Service	6
Figure 2: Instance Directory	6
Data Storage	6
Web Service	6
Windows Service	7
Instance Directory Utility	7
Policies	7
AlwaysChoice	7
ChoiceWhenNotExist	7
FailWhenNotExist	7
PreviousChoice	8
RestartWhenChanged	8
DirectDatabase	8
Default Instance	8
Command-line Arguments	9
Help	9
Output	9
Policies	9
Direct Database	9
Overriding policy behaviour	10
	2

Windows Services	10
Edit the registry	10
Use Service Control Manager	10
Process Runner	11
SETUP	11
TROUBLESHOOTING	11

SECTION 1 OVERVIEW AND PURPOSE

The Instance Directory is a directory of all Case Manager databases available to the Case Manager applications. Applications can detect the directory on the local network and connect to the appropriate database as defined by the policy set up in the application. It simplifies maintenance on a client site as connection management is centralised. Changes in database connection information is communicated across all workstations without any interaction with the actual workstation.

This document explains the technical working of the Instance Directory in all its components which will aid in implementation, maintenance and support.

SECTION 2 DEFINITIONS

2.1 Instance

An Instance is a named Case Manager database. It consists of a unique name by which it will be identified in the directory and the database connection details.

2.2 Default Instance

The Default Instance is a reference, stored per workstation (per user), to the instance the applications on that workstation should connect to by default. Only the applications that make use of policies dependent on the default instance will be affected by changes to the workstation's default instance value.

2.3 Instance Directory

The Instance Directory is a centralised directory of the available instances on the network. It is discoverable on the network; therefore, there should only be one Instance Directory installed on an entire network.

2.4 Instance Directory Windows Service

The Instance Directory operates via a Windows Service (typically installed on the server) which publishes the necessary Web Service and maintains the instance information.

2.5 Local Instance Directory

A Local Instance Directory is stored and maintained in the Windows Registry on the workstation per user. The Local Instance Directory is updated to contain the same information as the Instance Directory (on the server).

2.6 Instance Directory Utility

The Instance Directory Utility provides management of the Instance Directory and all the instance information. It also provides processes such as Backup and Restore capabilities, Database user verification, and Case Manager version upgrades.

2.7 Policy

To initialise the application to connect to a specific instance, an update, selection and failover process is needed. The combination of the three is defined in a *Database Connection Factory Initialisation Policy* or simply, *Policy*.

2.7.1 Update

The policy defines how the Local Instance Directory is updated to contain the same information as the Instance Directory installed on the server. This process is defined in how the application interacts with this process.

2.7.2 Selection

Secondly, the policy defines how an instance must be selected from the Local Instance Directory to initialise the current application database connection.

2.7.3 Fail-over Policy

A Fail-over process defines how the application should behave when the initialisation of the database connection fails.

SECTION 3 HOW DOES IT WORK?

3.1 Overview

VN Database

Figure 1: Database connection initialisation overview

3.1.1 Start-up

The Instance Directory is detected on the local network upon start-up of a Case Manager application. This functionality is made possible by a Web-Service Discovery mechanism called [WS-Discovery](#). If multiple Instance Directories exist on the local network, the first to respond will take preference. Of course, this can cause unpredictable behaviour and should be avoided.

3.1.2 Local Instance Directory Update

After detecting the Instance Directory on the local network, a local copy of the directory is updated. This copy (referred to as the Local Instance Directory) is stored per user on the current machine and contains all the information of the visible instances in the directory. Interaction, of the update process with the application starting up, is defined in the application's policy. Section 3.3 - [Instance Directory Utility](#) elaborates on the available policies.

3.1.3 Instance Selection

After the update of the Local Instance Directory, an instance is selected that will be used in the current application. The selection process is also defined in the application's policy and may involve user interaction (Section 3.3 - [Instance Directory Utility](#)). If the update process fails or was disabled in the application, selection will still be attempted from the instances as specified in the (possibly outdated) Local Instance Directory.

3.1.4 Database initialisation

Having the selected instance, the application can continue to initialise its database connection. This connection is subsequently tested which will mark the first interaction of the application with the database specified in the instance. The application may also include a database version check as part of this test.

3.1.5 Fail-over

It may be that the database connection test fails due to, among other things, incorrect details specified in the instance, network problems, and an incorrect application or database version. The policy of the application also specifies a Fail-over process which defines the behaviour of the application in this event. Depending on the policy, this may take the application back to a selection phase in the process.

3.1.6 Application

Control is finally given to the application itself, furnished with a working centralised-managed database connection.

3.2 Instance Directory Service

Figure 2: Instance Directory

The Instance Directory as installed (typically) on a server and consists of a locally stored XML data file containing the instance information exposed through a network-wide discoverable web service, governed by a windows service. Since the Instance Directory serves all Case Manager applications on the network with instance information, there should only be one Instance Directory installed on a network. Applications will connect to the first Instance Directory to respond and disregard the rest. Multiple Instance Directories might consequently produce unpredictable and confusing behaviour.

3.2.1 Data Storage

The Instance Directory does not rely on any database and is, therefore, extremely lightweight and easy to install. The Instance Directory data are stored in a local XML file which is located in the installation folder of the service. To secure access to the data this location should not be shared on the network; it should only be accessible through the web service.

3.2.2 Web Service

A discoverable Web Service is published and used by all Case Manager applications on the network to access instance information. Upon start-up of the Web Service, the current machine's IP address is detected and used to publish the Web Service. It is possible to override this behaviour by specifying an IP address in the service configuration file explicitly. The default port for the Web Service is **TCP port 3737**, which can also be set in the configuration file. Web Service discovery, however, requires an additional port to be open: **UDP port 3702**. Although direct access to the Web Service would in general not be required, it can be accessed via the following address:

<http://<ipaddress>:<port>/VoyagerNetz/InstanceDirectory/Service>

3.2.3 Windows Service

The Web Service is published and governed by a Windows Service that is easily installed through an installation wizard. The service is automatically started during installation and can be accessed through “Services” in Windows Control Panel. For the purpose of troubleshooting, events (such as errors that might occur in the Instance Directory service) are logged in the Event Logs of Windows with source reference “VN-InstanceDirectory”.

3.2.4 Instance Directory Utility

A locally Click-Once published utility is provided to set up and administer the Instance Directory. The utility can also detect multiple Instance Directories on a local network (which can cause some trouble in the application behaviour), and provides instance-related processes such as backup and restore, database user verification, and Case Manager version upgrades.

Refer to the Instance Directory Utility User Guide for more information.

3.1. Policies

The application policy (also referred to as the *Database Connection Factory Initialisation Policy*) specifies how the application should update the Local Instance Directory, select the appropriate instance, and behave when it fails to initialise. The following policies are available:

3.1.1. AlwaysChoice

The *AlwaysChoice* policy will always give the user the option to choose the appropriate instance when more than one is visible in the directory (when only one is available, that instance will be auto-selected). Upon start-up, the application will first update the Local Instance Directory to ensure the most recent instance information is available, and it will give the user the option to select another instance if it fails to initialise. Applications that use this policy by default are the Case Manager Configuration Utility and the Case Manager Wall Dashboard.

3.1.2. ChoiceWhenNotExist

Applications making use of the *ChoiceWhenNotExist* policy will update the Local Instance Directory upon start-up, after which the default instance (or a specific instance as specified by command-line argument) will be loaded. If the specified instance does not exist or the instance fails to load, the user will be presented with a choice of the available instances. Applications that make use of the *ChoiceWhenNotExist* policy are the Bulk Action Utility and the Content Manager.

3.1.3. FailWhenNotExist

The *FailWhenNotExist* policy operates similarly to the *ChoiceWhenNotExist* policy in that the Local Instance Directory is updated upon start-up and the default

instance or a specified instance is used. The user, however, is never presented with a choice: the application fails when the instance does not exist or if it fails to load. For scheduled and automated processes this is generally the best policy to use. Applications that operate on the *FailWhenNotExist* policy are the Daily Maintenance Runner, the Process Runner, the SMS Merger, and the Windows Services for the Time- and Queue Servers.

3.1.4. PreviousChoice

The *PreviousChoice* policy will use the last successful configuration (instance) to initialise the connection of the application. After the update of the Local Instance Directory, the application will communicate to the user an option to reconfigure with a time-out of 5 seconds. If the user chooses to reconfigure, an option of all available instances is presented. Otherwise, the previous configuration is used. If the application fails to initialise with the relevant instance, a choice is also presented. The Time- and Queue Server Console Applications make use of the *PreviousChoice* policy.

3.1.5. RestartWhenChanged

Similar to the *ChoiceWhenNotExist* policy, the *RestartWhenChanged* policy makes use of the default instance (or specific instance as specified as a command-line argument) and presents a choice of instances upon failure. The difference lies in the update of the Local Instance Directory. If the Local Instance Directory information is available, the initialisation will take place concurrently with the update of the Local Instance Directory. If the updated information differs to the instance information used during the initialisation, the application is restarted. This policy is only supported in applications that support restart (which depends on the implementation of the application). The Case Manager Windows Application makes use of this policy.

3.1.6. DirectDatabase

DirectDatabase is not strictly a policy as instance information is not used. It provides a mechanism of specifying the database connection details directly without making use of instances or the Instance Directory. The Local Instance Directory is not updated and the application will fail if the database connection initialisation fails.

3.2. Default Instance

The Default Instance is the name of the instance to be used by default and is stored in the Local Instance Directory per user (and per machine, as it is stored in the Local Instance Directory). Applications that make use of policies such as *RestartWhenChanged*, *ChoiceWhenNotExist* and *FailWhenNotExist* can make use of the Default Instance to initialise the application without prompting the user for the choice of instance.

The Default Instance is updated when the database connection details are changed in one of these applications. For example, when “Configure Database Connection Settings” is selected on the login form of Case Manager, the user is prompted to

select a new instance to connect to. The instance will be stored as the Default Instance and will affect all applications dependent on the Default Instance.

3.3. Command-line Arguments

All applications integrated with the Instance Directory support command-line arguments to override the default behaviour in relation to the initialisation process. This is especially handy when automated applications are scheduled. The following command-line options are defined in all Instance Directory integrated applications:

3.3.1. Help

More information on the supported command-line arguments can be found by running the application with the `-help` argument. This will guide the user on which options are available. The help option will override all other options and application behaviour and will only display the help.

3.3.2. Output

The `-output` argument is used to specify how initialisation error messages should be displayed. The available options are:

- **Console:** All error messages are displayed in the console from where the application was started. This option can be used when error messages are redirected to file. For example:

```
application.exe -output=Console > error.log
```

- **Application:** Error messages are displayed through the application. If the application is a windows application, an error message box would be used.

3.3.3. Policies

The application behaviour, in terms of the policies discussed in Section 3.4 - [Policies](#), can be specified in the application by making use of the `-policy` option. For example:

```
application.exe -policy=AlwaysChoice
```

Furthermore, for policies that support specified instances the following two options may be used:

- **DefaultInstance** forces the application to make use of the default instance of the current user on the current workstation.
- **Instance** allows you to specify the instance to use by name. for example:

```
application.exe -policy=FailWhenNotExist  
-instance="Test database"
```

3.3.4. Direct Database

When an application should specify the database details directly without making use of the Instance Directory, the `DirectDatabase` option may be used to specify the following four parameters in a semicolon-separated list:

- **DataSource** specifies the server name or address.

- **Port** specifies the port that should be used. The default value is 3050.
- **Path** specifies the directory on the server where the database resides.
- **Database** specifies the name of the database file.

For example:

```
application.exe -Direct="DataSource=127.0.0.1;
    Port=3050;Path=C:\;Database=casemanager.fdb"
```

3.3.5. Overriding policy behaviour

It is possible to make use of the policies as discussed so far without updating the Local Instance Directory and without the Fail-over process, by simply adding these options:

- `-NoUpdate` will not update the Instance Directory.
- `-NoFailOver` will fail the application when a problem occurs during initialisation.

3.3.6. Windows Services

Command-line arguments can be used with Windows Services, however, this requires a bit more setup. The actual program that is executed when a service starts is specified in the registry; this is where we have to add the arguments. Typically this is done when one does not want to rely on the Default Instance (which might change) for the appropriate service, or when the servers of different instances are run on one machine.

There are two ways to change the path executed when the service starts up:

3.3.6.1. Edit the registry

1. Run `regedit`
2. Go to:
`HKLM\SYSTEM\CurrentControlSet\Service\<ServiceName>`
 with `<ServiceName>` the name used for the service when viewed in Windows Control Panel.
3. Find the property `ImagePath`
4. Add appropriate arguments to the path

3.3.6.2. Use Service Control Manager

1. Open Console window in Administrator mode
2. Run the command
`sc config <ServiceName> binPath=<newPath>`
 Where
`<ServiceName>` should be the name of the service as used by Windows (use quotation marks if it contains spaces).
`<newPath>` should be the new full path to the executable including the appropriate command-line arguments. Make use of quotation marks.

3.4. Process Runner

The Process Runner is designed to run instance processes as a schedulable program. The only process currently supported by the Process Runner is the Backup process. As the application is integrated with the Instance Directory, all the command-line arguments discussed above apply. In addition to that, the arguments should specify the process and the arguments related to the process. To view help on the Backup process, run the Process Runner with the following arguments: `-process=BackupInstance -help`

The following three arguments, as mentioned in the help, can be specified for the BackupInstance process:

BackupDirectory: the directory where the backup made should be stored.

LogDirectory: the directory where the backup log should be stored.

GarbageCollection: `true` or `false`, whether or not garbage collection should be done.

SECTION 4 SETUP

Installing the Instance Directory is very simple. No further setup is required after installation other than ensuring the necessary ports are open. For security purposes, it is strongly recommended not to use a network-shared location for the installation directory. The Instance Directory will detect the local-network address on the machine it is running on and publish the web service on port 3737. If necessary, the Instance Directory configuration file (in the installation directory of the Instance Directory) can be used to set the specific address and port that it should be published on. To set the port, simply set the `PortNumber` value in the XML Configuration file. To set the address to a specific IP Address or DNS Name, set `AutoDetectNetwork` to `false` and `FixedIPAddress` to the appropriate value.

SECTION 5 TROUBLESHOOTING

<i>Problem</i>	<i>Tests & Possible Solutions</i>
Forgot Instance Directory Utility password	Edit the <code>VoyagerNetzInstanceDirectory.xml</code> file in the installation folder of the Instance Directory. Open the utility and
	Remove the entire <code>Security</code> section. This will reset the password to the default "badpassword".
	Open the utility and change the password to appropriate password.
<i>Directory not detected</i>	Make sure the instance directory is installed and service is running.
	Check the configuration of the Instance Directory service to ensure that auto detect network is true or that the correct address that translates to the current server is set.
	Check that UDP port 3702 and TCP port 3737 (or the port that was set in the configuration file) is open on the server.
	Open the web service page in the browser (on the server and then on the client) to identify if a network problem exists.

	Open the Instance Directory Utility on the client machine to see if directory is detected via a different application.
<i>No instances detected</i>	Open Instance Directory Utility and verify that visible instances are available.
<i>Instance information incorrect</i>	Open Instance Directory Utility and verify that only Instance Directory exists on the current local network.
	It might be that the Local Instance Directory is not updated and out-of-date information is used to initialise the application. Verify that you can connect to the Instance Directory and that the application does not specify that no update of the directory should be done.

This list is by no means complete. When you encounter a common problem, please let us know so that we can grow the list of troubleshooting tips.