FS – External Component Register

Content

1 INTRODUCTION 2

2 FUNCTION REQUIREMENTS (COMMERCIAL) 2

3 FUNCTION SPECIFICATION (DESIGN RELATED) 2

3.1 INTRODUCTION 2
3.2 EXPORTED INTERFACES 2
3.2.1 ILocateService 2
3.2.2 IRegisterService 3
3.2.3 IServiceInstance 3
3.3 IMPORTED INTERFACES 4
3.4 FUNCTIONS 4
3.4.1 LDAP Communication with MCR 4
3.4.2 Locate a Service 4
3.4.3 Configure a Service 6
3.4.4 Register a Service 7
3.4.5 Configuration of Available Service Properties 7
3.4.6 Events 8
3.4.7 Thread Safe 8

4 EXTERNAL OPERATION CONDITIONS 8

4.1 CONFIGURATION 8
4.2 LOGGING 8

5 CAPABILITIES 9

6 REFERENCES 9

7 TERMINOLOGY 9

History

Version	Date	Adjustments
PA1	2007-08-28	First version of the document. Original document was found in MAS. (EMAHAGL)
PA2	2007-09-26	Updated with service status functionality (EMAHAGL)

PA3	2008-03-11	Updated with command line functionality (EERITOR).

1 Introduction

This document specifies the function of the External Component Register component. The External Component Register provides an interface that can be used by other components in order to retrieve a service instance that can be used for communication with other parts of the M3 system.

2 Function Requirements (Commercial)

This paragraph is intentionally left blank.

3 Function Specification (Design Related)

3.1 Introduction

The External Component Register handles the communication with MCR. The External Component Register can be used to register a service in MCR or to lookup services already registered in MCR.

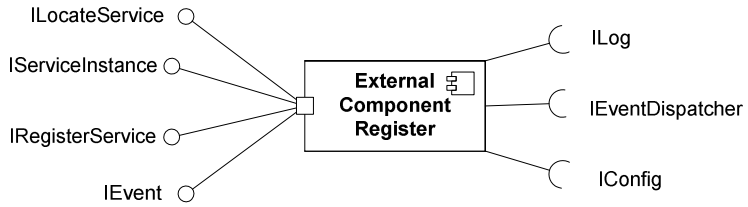

Figure 1 External Component Register component

3.2 Exported Interfaces

3.2.1 ILocateService

This interface is used to lookup registered components.

3.2.1.1 Methods

ILocateService offers the following methods:

- *IServiceInstance locateService(String serviceName)*
  Locates one instance that runs the specific service (as described in 3.4.1).
- *IServiceInstance locateService(String hostName, String serviceName)*
  Locates one instance having the host name hostName that runs the specific service (as described in 3.4.1).

- *IServiceInstance getAnotherService(IServiceInstance service)*
  Used by clients to indicate that another service instance is desired instead of the given one. This method is used when a previously located service is not sufficient, but there is no actual error to report.
- *void reportServiceError(IServiceInstance service)*
  Used by clients to report that an error occurred when communicating with the service.
- *List<IServiceInstance> getServiceInstances(String serviceName)*
  Retrieves a list of all ServiceInstance objects that are used for the specific service (as described in 3.4.1).

3.2.2 IRegisterService

This interface is used to register a service in MCR.

The External Component Register also provides a command line interface to the IRegisterService services in MCR making it possible to access registerService and unregisterService from a shellscript.

3.2.2.1 Methods

IRegisterService offers the following methods:

- *void registerService(IServiceInstance service)*
  Registers the given service in MCR.
- *void unregisterService(IServiceInstance service)*
  Unregisters the given service in MCR.

3.2.3 IServiceInstance

This interface is used to access properties of a service instance in MCR.

3.2.3.1 Classes

The enum *ServiceStatus* is used to indicate the status on the service. It can be UP or DOWN.

3.2.3.2 Methods

IServiceInstance offers the following methods:

- *String getProperty(String propertyName)*
  Returns the requested property. For available properties see [2].
- *void setProperty(String propertyName, Object propertyValue)*
  Sets the specified property. For available properties see [2].
- *void setServiceStatus(ServiceStatus serviceStatus)*
  Sets status on the service instance
- *ServiceStatus getServiceStatus()*
  Retrieves status on the service instance

3.3 Imported Interfaces

The External Component Register uses the following external interfaces:

- *ILog* for logging
- *IConfig* for configuration
- *IEventDispatcher* for retrieving events

3.4 Functions

3.4.1 LDAP Communication with MCR

The External Component Register communicates with MCR using LDAP. An internal interface IDirectoryService is used towards the LDAP implementation.

3.4.2 Locate a Service

The External Component Register can be used to locate a service. Figure 2 exemplifies how a service is located.

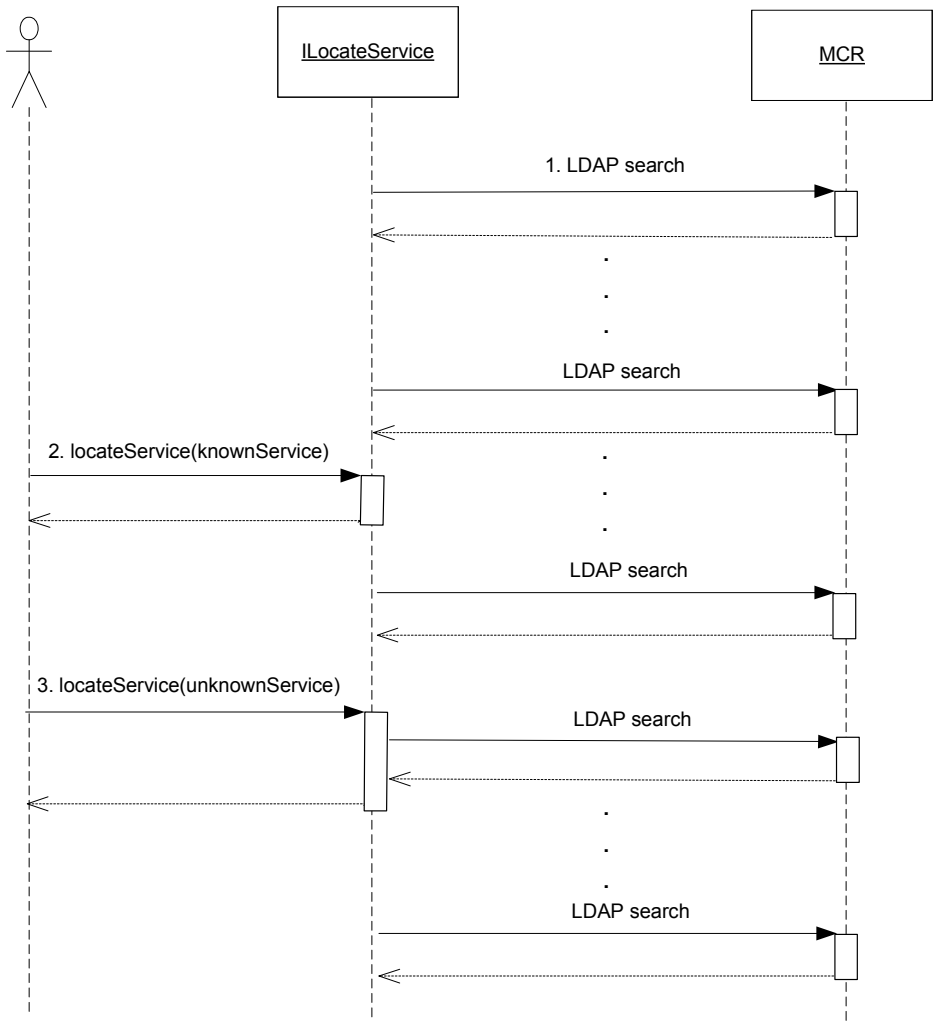

Figure 2 Example of locating a service

1. Lookup in MCR is performed at a regular basis.
2. A known service is requested using the *ILocateService* interface. Since it has already been looked up it can be returned immediately.
3. An unknown service must be looked up in MCR before returning.

The External Component Register has prior knowledge of services that has been configured. Other services become known to External Component Register when first asked for.

When a client wishes to locate a known service, External Component Register returns immediately with an appropriate service instance. For an unknown service, External Component Register first needs to lookup the service in MCR, after which the service is considered known.

Failure to contact the MCR is logged as an error.

If an unknown service is requested and not found in MCR the failure is propagated to the user of the External Component Register.

3.4.2.1 Periodical Update

Services known by External Component Register are looked up periodically in MCR. The periodicity of the lookup is configurable.

3.4.2.2 MER Algorithm

An appropriate service instance is located using the MER algorithm, see [1].

3.4.2.3 Other Algorithms

It is possible to configure the use of other algorithms than the MER algorithm for locating service instances for a specific service.

The External Component Register supports the additional algorithm for handling of logical hosts and multimaster environments for UserRegister service instances as described in [3].

3.4.2.4 Detection of Removed Service instances

During a periodical update of services, the External Component Register may detect that a previously known service instance has disappeared. In this situation, the External Component Register generates the event *Service Not Available*.

3.4.2.5 Report Service Instance Error

When a client has requested a service that does not seem to work, this is reported to the External Component Register using the method *reportServiceError*. The status is set to DOWN.

If an unknown service is reported the failure is propagated to the user of the External Component Register.

3.4.3 Configure a Service

The External Component Register can be configured with the services that shall be used by the system. Although a service is not configured, it can still be located by the External Component Register. However, if a service is configured features such as configured service instances, default properties (see 3.4.3.1) and Override MCR (see 3.4.3.2) can be used.

3.4.3.1 Configured service instances

For each configured service, it is possible to specify configured service instances to use in case MCR is not reachable or has no resources for the service in question. If no service instance is configured and the service is not found in MCR, the service is considered not available and will be propagated to the user of the External Component Register.

It is also possible to configure default properties for a configured service. This can for example be a default port number to use in case the service instance defined in MCR is missing this property.

3.4.3.2 Override MCR

For each configured service, it is possible to specify that MCR should be overridden. In that case only configured instances are used instead.

3.4.4 Register a Service

The External Component Register can be used to register a new service instance. Figure 3 exemplifies how a service instance is registered.

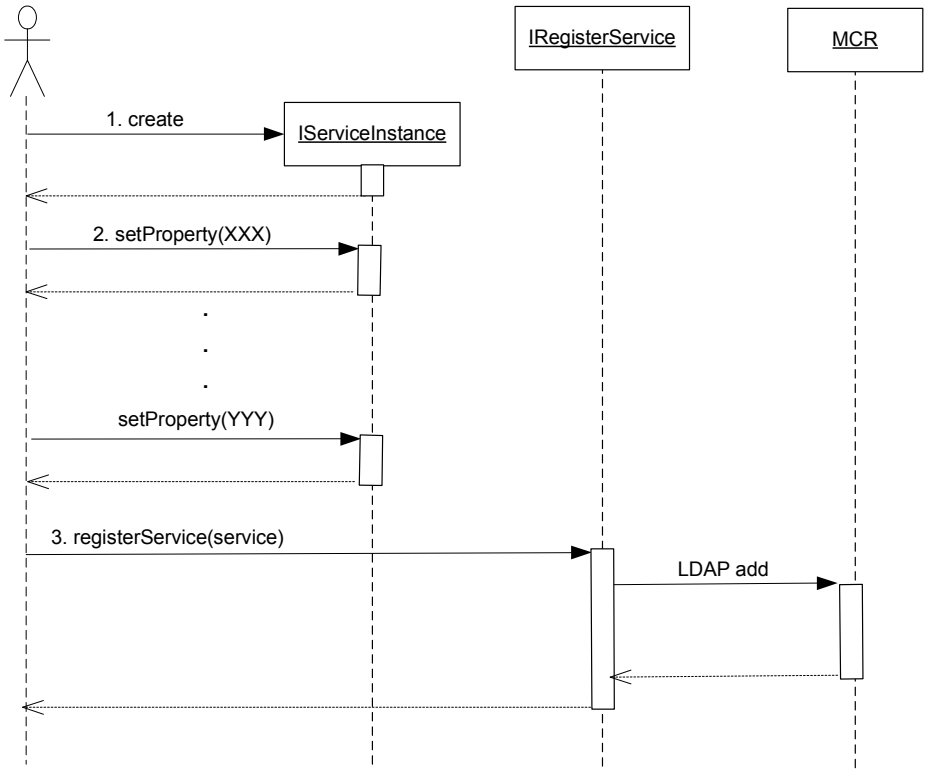

Figure 3 Example of registering a service instance

1. A new service instance is created.
2. The service instance properties are set.
3. The instance is registered in MCR using the *IRegisterService* interface.

If the service could not be registered due to communication problems or problems with the host (e.g. write protection) this is logged as an error and propagated to the user of the External Component Register.

3.4.5 Configuration of Available Service Properties

The possible properties of a service are described in [2].

The properties that should be available in a located service (and thus retrieved from MCR) are configurable in the External Component Register.

If a new service property is introduced in the M3 system, it will be possible to use that property (when locating a service or register a new service) simply by modifying the External Component Register configuration.

A request for an unknown service property is not logged but propagated to the user of the External Component Register.

3.4.6 Events

3.4.6.1 Consumed

The External Component Register registers itself using the *IEventDispatcher* interface as a receiver of the following events:

- Configuration has changed (indicates that configuration should be re-read)

3.4.6.2 Produced

The External Component Register generates the following events:

- Service Not Available
  When a previously known service has disappeared from MCR. The event also contains the name of the unavailable service.

3.4.7 Thread Safe

The External Component Register is thread safe.

4 External Operation Conditions

4.1 Configuration

The following are configurable in the External Component Register:

- Timeout of LDAP command
- Amount of retries for an LDAP command
- Services, possibly with configured instances and default properties and/or Override MCR feature
- Service properties that shall be possible to retrieve and set
- Location of MCR
- MCR lookup periodicity

4.2 Logging

- A host error is logged as an error only once when it occurs and once as a warning when the host is working again.
- Invalid configuration parameters are logged as a warning together with relevant default values.

- All calls to interface methods are logged at info level. Method name and parameter values are logged. At return returned value is logged.

5 Capabilities

This paragraph is intentionally left blank.

6 References

[1] MER Developer's Guide
1/1551–CRH 109 089 Uen

[2] MCR Developer's Guide
3/1551-CRH 109 581/1 Uen

[3] MUR Developers Guide
3/1551-CRH 109 086 Uen

7 Terminology

MCR	Messaging Component Register
MER	Message Event Repository