On the fly

User Guide

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Type of Document** | User Guide | | | | |
| **Reference:** |  | | | | |
| **Issue:** |  | **Revision:** |  | **Status:** | Draft |
| **Created by:** | Gino Mascotti | | | **Date:** | 30/09/2014 |
| **Updated by:** | Gino Mascotti | | |  | 19/12/2014 |
| **Approved by:** |  | | | | |

Summary

[**Project description** 4](#_Toc407620737)

[Features 4](#_Toc407620738)

[2.0 WSDL Operations 4](#_Toc407620739)

[2.1 WSDL Operations 4](#_Toc407620740)

[REST Operations 6](#_Toc407620741)

[How to retrieve the data 8](#_Toc407620742)

[2.1 OnTheFly 1.0 8](#_Toc407620743)

[2.2 OnTheFly 2.0 9](#_Toc407620744)

[Presentation 10](#_Toc407620745)

[**Configuration File** 11](#_Toc407620746)

[Global Settings 11](#_Toc407620747)

[Mapping Settings 13](#_Toc407620748)

[The Store Procedure 15](#_Toc407620749)

[**Error Management** 20](#_Toc407620750)

[Fault Error 20](#_Toc407620751)

[**Attributes management** 22](#_Toc407620752)

[Frequency 22](#_Toc407620753)

[**Log Management** 23](#_Toc407620754)

# **Project description**

The project OnTheFly WebServices was created to execute queries SDMX to different types of databases.  
Is no longer necessary to have the database structured according to a specific model but simply the presence of some store procedures to perform all queries and create "On the Fly" all required structures.

## Features

OnTheFly WebServices answers all queries Soap or Rest both in SDMX v2.0 that SDMX v2.1  
Functionality implemetate are:

### 2.0 WSDL Operations

Data

|  |  |
| --- | --- |
| WebMethod | Status |
| GetGenericData | Supported |
| GetCompactData | Supported |
| GetUtilityData | Not planned |
| GetCrossSectionalData | Not planned |

Registry-Oriented Web Service Functions to get Structural metadata

|  |  |
| --- | --- |
| WebMethod | Status |
| QueryStructure | Supported |

QueryStructure queriable artefacts

* Dataflow
* Codelist
* ConceptScheme
* CategoryScheme
* KeyFamily
* AgencySchema

### 2.1 WSDL Operations

|  |  |
| --- | --- |
| WebMethod | Status |
| Data |  |
| GetStructureSpecificData | Not planned |
| GetGenericData | Not planned |
| GetStructureSpecificTimeSeriesData | Supported |
| GetGenericTimeSeriesData | Supported |
| Structure usage |  |
| GetDataflow | Supported |
| Structure |  |
| GetDataStructure | Supported |
| Item scheme |  |
| GetCategoryScheme | Supported |
| GetConceptScheme | Supported |
| GetCodelist | Supported |
| GetHierarchicalCodelist | Not planned |
| GetOrganisationScheme | Supported |
| GetReportingTaxonomy | Not planned |
| Other maintainable artefacts |  |
| GetStructureSet | Not planned |
| GetProcess | Not planned |
| GetCategorisation | Supported |
| GetProvisionAgreement | Not planned |
| GetConstraint (content) | Not yet implemented |
| XML Schemas (XSD) |  |
| GetDataSchema | Not planned |
| Generic query for structural metadata |  |
| GetStructures | Not planned |

### REST Operations

Structural Metadata Queries

|  |  |
| --- | --- |
| Resources defined | Status |
| datastructure | Supported |
| categoryscheme | Supported |
| conceptscheme | Supported |
| codelist | Supported |
| hierarchicalcodelist | Not planned |
| organisationscheme | Supported |
| agencyscheme | Supported |
| dataproviderscheme | Not planned |
| dataconsumerscheme | Not planned |
| organisationunitscheme | Not planned |
| dataflow | Supported |
| reportingtaxonomy | Not planned |
| provisionagreement | Not planned |
| structureset | Not planned |
| process | Not planned |
| categorisation | Supported |
| contentconstraint | Not planned |
| attachmentconstraint | Not planned |
| structure | Not planned |

## How to retrieve the data

Currently the project has two versions OnTheFly:

### 2.1 OnTheFly 1.0

In this version uses a single database is to retrieve data and metadata. Now let's see in detail how the metadata are taken

* **Category and Categorisation**

This metadata can be retrieved by taking information from another database that will have a table "ContentTree". This table contains information about the themes and the masts to be used. In the CategoryName in configuration parameter CategorySetting, you can enter the name in the theme to use if they are more than one.

Alternatively you can use a StoreProcedure that return directly the necessary information to the population of these metadata.

* **Dataflows**

The dataflow is retrieved through the store procedures GetDatasets.  
These will be taken as the key to building all other artifacts.

* **DataStructures**

The dsd are created on the fly for giving the premise that for each dataset there is one and only one dsd. The code is created with the code of the dataflow and the configuration parameter DsdFormat description will be used for the description of the Dataflow.  
For each Dsd is always only one ConceptScheme whose concepts will be taken as described below

* **ConceptScheme**

Also the ConceptScheme are created on the fly, one for each dsd. The code consists of the code of Dataflow with the configuration parameter ConceptSchemeFormat description will be used for the description of dsd

The concepts will be so derived:

For dimensions using StoreProcedure: GetDimensions

For attributes or using StoreProcedure:GetAttributes or these will be read from a file that is in ConfigurationXml\ AttributeConcepts.xml

Will be added an other attribute FLAG if you set the configuration parameter: ConceptObservationFlag

Will be added automatically the OBS\_VALUE primary measures

* **Codelist**

For codelist concepts of type dimension, we can derive codelist or constrain (related Dataflow) with StoreProcedure: GetDimensionCodelistConstrain or not constrain going to call the same StoreProcedure for all Dataflow found or send the system in error by setting the configuration parameter CodelistWhitoutConstrain to false.

For codelist concepts of type attribute, we can derive codelist constrain (related Dataflow) with StoreProcedure: GetAttributeCodelistConstrain and not constrain the StoreProcedure: GetAttributeCodelistNOConstrain oppure il sistema andra a prendere le codelist dal file che si trova in ConfigurationXml\ AttributeConcepts.xml

For codelist concepts of frequency can not find the concept associated with the dataflow but since this is required to build the dsd, in this case we're going to take the codelist in the file that is located in ConfigurationXml\FrequencyCodelist.xml. In the case the concept is the codelist will be taken as any other Dimension

### 2.2 OnTheFly 2.0

In this version is used database for data and another for Metadata with the exception that I report later. To upgrade to OnTheFly 2.0 version you must enter the connectionstring in the second database configuration parameter MsConnectionString

* **Category and Categorisation**

This metadata will be retrieved in the same way as the OnTheFly 1.0 version

* **Dataflows**

The dataflow is retrieved through StoreProcedure: GetDataflows

* **DataStructures**

The dsd are recovered through StoreProcedure: MSGetDSD

* **ConceptScheme**

The ConceptScheme are recovered through StoreProcedure: MSGetConceptScheme

* **Codelist**

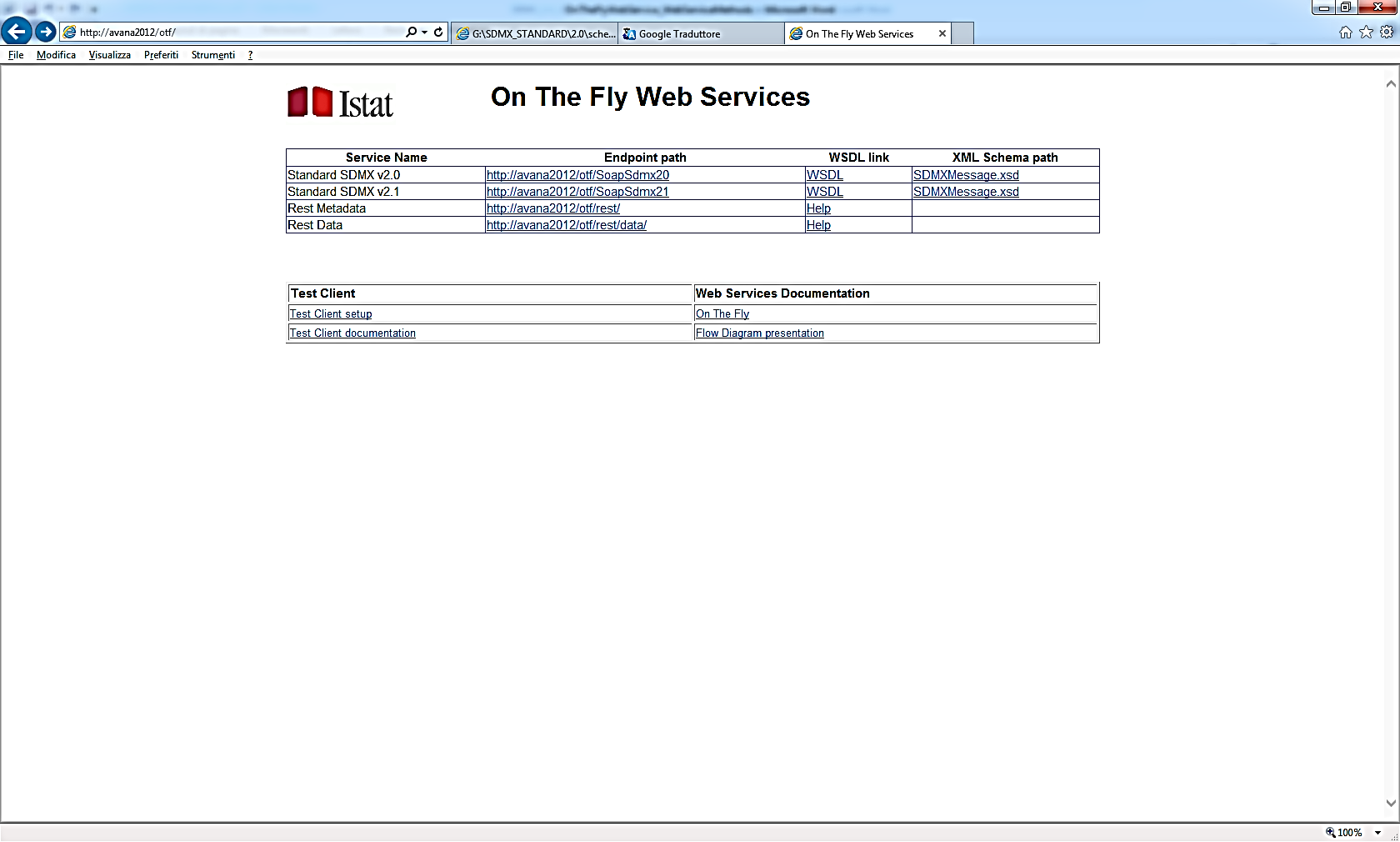
The codelist not constrinate are recovered through StoreProcedure:

MSGetCodelist

For codelist constrainate instead it all use the same methods and mechanisms used in version 1.0 OnTheFly

## Presentation

Browsing to where you installed the service will display the welcome page:



On this page you can view the addresses of the application and download the test client and manuals

# **Configuration File**

The configuration file of the project OnTheFly WebServices

**bin\ServiceConfiguration.xml**

This file is composed of two sections

* **Global Setting**
* **Mapping Settings**

## Global Settings

In this section you will find all the configuration parameters required and used in all versions dell'OnTheFly  
The parameters are composed of

<add key="[key]" value="[value]" />

|  |  |
| --- | --- |
| Key | Description |
| MainAgencyId | AgencyScheme Structure and descriptions  Is posible insert in this tag one or more name with this sintax:  <Name LocaleIsoCode="en"><![CDATA[Agency Name]]></Name> |
| OrganisationScheme | OrganisationScheme Structure and descriptions  Is posible insert in this tag one or more name with this sintax:  <Name LocaleIsoCode="en"><![CDATA[ISTAT AGENCIES]]></Name> |
| Version | Default Version of artefacts |
| DatasetTitle | Is a true/false value and determinate if you want show a Dataset Title in data query result at Dataset Level. It will taken the first description available |
| UserName | This parameter is required by some Store procedure |
| Domain | This parameter is required by some Store procedure |
| LogLocation | Determinate a location of application Log. Is possible use a special Word as %Temp% for identify a temporary Path |
| LogLevel | Possible value:  **"None"** -> No log  **"All"** -> All log (Debug Mode)  **"Warning**" -> Only Warning and Error Log  **"Error"** -> Only Error Log |
| HeaderSettings | Is possible configure all Header Settings that you want to return the header of each query |

## Mapping Settings

In this section you can save one or more possible configurations depending on the version you want to use.

In the main node MappingSettings is an attribute UsedSettings that should be enhanced with the name of the configuration you want to use

To create a new configuration must

<MappingSetting id="SettingsId">

The id is the name you want to give to the configuration.

Possible parameters:

|  |  |
| --- | --- |
| Key | Description |
| ConnectionString | ConnectionString for connection to SqlServer Database.  In this Database must be present all Store Procedure under described |
| MsConnectionString | ConnectionString for connection to MappingStore Database  In this Database must be present all Store Procedure under described. If this is valorizated the versione of program automatically change in 2.0 |
| CodelistWhitoutConstrain | Set to true for merging codelist of all Dataflow (low performance)  when arrive Query Codelist whitout Constrain (Dataflow reference)  otherwise an exception will be generated. |
| ConceptObservationFlag | You can put the FLAG concept in dsd, in the value field must be specified ID in the attachmentLevel field the level where the data should be inserted inside, assignmentStatus if it is mandatory or optional and descriptions names.  The codelist of ConceptObservationFlag will be taken to the store procedure “GetFlags”.  If you do not want to add this attribute to remove the tag from the configuration file. |

|  |  |
| --- | --- |
| CategorySettings |  |
| ConnectionStringCategory | ConnectionString Alternative for retreival CategoryScheme, Category, Categorisation  In this Database must be present a table ContentTree |
| CategoryName | Category name you want to use in ContentTree Table |

The formats of the id of the artifacts made

|  |  |
| --- | --- |
| Key | Description |
| DsdFormat | Format of KeyFamily and DataStructure ID. The characters {0} will be sobstituted with Dataflow ID |
| ConceptSchemeFormat | Format of ConceptScheme ID. The characters {0} will be sobstituted with Dataflow ID |
| CodelistFormat | Format of Codelist ID. The characters {0} will be sobstituted with Concept ID |
| CategorySchemeFormat | Format of CategoryScheme ID. The characters {0} will be sobstituted with Dataflow ID |
| CategorisationFormat | Format of Categorisation ID. The characters {0} will be sobstituted with Dataflow ID |

### The Store Procedure

|  |  |
| --- | --- |
| Key | Description |
| GetCategory | Get Category and Categorisation, if this not exist and is populate CategorySettings, the OnTheFly system get Category information from other Database  *Parameter:*  UserName, User that use StoreProcedure  Domain: user domain  TimeStamp: date for require data from last update |
| GetDatasets | Get Datasets, required in OTF 1.0  *Parameter:*  UserName, User that use StoreProcedure  Domain: user domain  TimeStamp: date for require data from last update |
| GetDimensions | Get Dimensions,required in OTF 1.0  *Parameter:*  Code: Dataset Code  TimeStamp: date for require data from last update |
| GetDimensionCodelistConstrain | Get dimension’s codelist constrained (only used in Dataset specified), required in OTF 1.0  *Parameter:*  DatasetCode, Dataset Code  DimCode: Dimension Code  UserName, User that use StoreProcedure  Domain: user domain  TimeStamp: date for require data from last update |
| GetDimensionCodelistNOConstrain | Get dimension’s codelist not constrain, if this not exist, the OnTheFly system check the flag “CodelistWhitoutConstrain”  *Parameter:*  DimCode: Dimension Code  Depth, depth of the level of codelist  UserName, User that use StoreProcedure  Domain: user domain  TimeStamp: date for require data from last update |
| GetAttributes | Get Attributes, if this not exist, the OnTheFly system get a Attribute from File “Attributes.xml”  *Parameter:*  Code: Dataset Code  TimeStamp: date for require data from last update |
| GetAttributeCodelistConstrain | Get Attribute’s codelist constrained (only used in Dataset specified), if this not exist, the OnTheFly system get a Attribute from File “Attributes.xml”  *Parameter:*  DatasetCode, Dataset Code  AttributeCode: Attribute Code  Depth, depth of the level of codelist  UserName, User that use StoreProcedure  Domain: user domain  TimeStamp: date for require data from last update |
| GetAttributeCodelistNOConstrain | Get Attribute’s codelist not constrained, if this not exist, the OnTheFly system get a Attribute from File “Attributes.xml”  *Parameter:*  AttributeCode: Attribute Code  Depth, depth of the level of codelist  UserName, User that use StoreProcedure  Domain: user domain  TimeStamp: date for require data from last update |
| GetData | Get Data, ever required  *Parameter:*  DatasetCode, Dataset Code  WhereStatement: where condition each parameter in this format: [$parameter$].value  Time, where condition for time parameter  DataSetAction, Dataset action  UserName, User that use StoreProcedure  Domain: user domain  TimeStamp: date for require data from last update |
| GetGroups | Get Groups, only used in OnTheFly v2.0  *Parameter:*  DatasetCode, Dataset Code  Columns, each required item separated by Comma  UserName, User that use StoreProcedure  Domain: user domain  TimeStamp: date for require data from last update |
| GetDataflows | Get Dataflows, only used in OnTheFly v2.0  *Parameter:*  UserName, User that use StoreProcedure  Domain: user domain  TimeStamp: date for require data from last update |
| MSGetCodelist | Get Codelists, only used in OnTheFly v2.0  *Parameter:*  CodelistCode: Filter for Codelist Code  DSDCode, Filter for DSD Code  ConceptSchemeCode, Filter for ConceptScheme Code  AgencyId: structure Agency Id, Null for all  Version: structure version, Null for all  IsStub, if 1 return a stub structure  UserName, User that use StoreProcedure  Domain: user domain  TimeStamp: date for require data from last update |
| MSGetConceptScheme | Get ConceptSchemes, only used in OnTheFly v2.0  *Parameter:*  ConceptSchemeCode, Filter for ConceptScheme Code  DSDCode, Filter for DSD Code  CodelistCode: Filter for Codelist Code  AgencyId: structure Agency Id, Null for all  Version: structure version, Null for all  IsStub, if 1 return a stub structure  UserName, User that use StoreProcedure  Domain: user domain  TimeStamp: date for require data from last update |
| MSGetDSD | Get DataStructures, only used in OnTheFly v2.0  *Parameter:*  DSDCode, Filter for DSD Code  CodelistCode: Filter for Codelist Code  ConceptSchemeCode, Filter for ConceptScheme Code  DFId, Filter for Dataflow ID  AgencyId: structure Agency Id, Null for all  Version: structure version, Null for all  IsStub, if 1 return a stub structure  UserName, User that use StoreProcedure  Domain: user domain  TimeStamp: date for require data from last update |

# **Error Management**

The project OnTheFly in case of error returns a structure SDMX Standard for both errors resulting from calls Soap and for those calls from Rest.

**<s:Fault**>

<**faultcode**>s:500<**/faultcode**>

**<faultstring xml:lang**="it-IT">Internal Server Error<**/faultstring**>

<**faultactor**>GetGenericData<**/faultactor**>

<**detail**>

**<Error Type**="InternalError">

<**Message**>"WS Internal Error<**/Message**>

<**MessageDetail**>Dsd Not found for Dataflow code: TestDataFLOW<**/MessageDetail**>

<**Source**>FlyEngine.Model.RetrievalManager<**/Source**>

<**/Error**>

<**/detail**>

<**/s:Fault**>

The errors can be divided into two types:

* "Derived from the project"

They are all design errors, configuration or data retrieval from the database

* "Derived from CommonApi"  
  Are all errors that are triggered by commonApi as parse query or immutable instanced of the Sdmx Objects

In either case the system will return:  
FaultCode and faultString all possible configurations are described below  
faultactor, describes the entrypoint of the application that was called  
detail  
In Error is the type attribute that indicates whether the error is "Derived from the project" or "Derived from CommonApi"  
Message and MessageDetail are descriptions of the error  
Source is the name of the source object that triggered the error

For all errors "derivatives from the project" you can change the description and FaultCode editing the xml file that is located in

**“Bin/ConfigurationXml/ErrorDescription.xml”**

## Fault Error

|  |  |  |
| --- | --- | --- |
| Code | SdmxError | Description |
| 100 | NoResultsFound | No Results Found |
| 110 | Unauthorised | Unauthorised |
| 130 | ResponseTooLarge | Response Too Large |
| 140 | SyntaxError | Syntax Error |
| 150 | SemanticError | Semantic Error |
| 500 | InternalServerError | Internal Server Error |
| 501 | NotImplemented | Not Implemented |
| 503 | ServiceUnavailable | Service Unavailable |
| 510 | ResponseSizeExceedsServiceLimit | Response Size Exceeds Service Limit |

More information you can find in the manual SDMX WebServicesGuidelines

# **Attributes management**

The project OnTheFly has A supported feature that allows you to insert the attributes and their codelist even without of StoreProcedure dealing retrieve this information.

The system will retrieve the attributes and their codelist from a file that is located in

**“Bin/ConfigurationXml/AttributeConcepts.xml”**

In this file you can assign each dataflow attributes with their codelist in this way:

<Attribute Code="AVAILABILITY" attachmentLevel="Dataset" assignmentStatus="Mandatory">

<Name LocaleIsoCode="en"><![CDATA[Attribute AVAILABILITY]]></Name>

<Name LocaleIsoCode="fr"><![CDATA[Attribute AVAILABILITY]]></Name>

<Codelist>

<Code value="0">

<Name LocaleIsoCode="en"><![CDATA[Available]]></Name>

<Name LocaleIsoCode="fr"><![CDATA[Available]]></Name>

</Code>

<Code value="1">

<Name LocaleIsoCode="en"><![CDATA[not Available]]></Name>

<Name LocaleIsoCode="fr"><![CDATA[not Available]]></Name>

</Code>

</Codelist>

</Attribute>

How can see, we have defined an attribute by Code and Name nodes. Attributes that are required: attachmentLevel describe where they will be placed within the attributes of the data structure and can be:

Dataset

Observation

DimensionGroup (at the level of the series)

And assignmentStatus describes whether the attribute is Mandatory or Conditional

In node Codelist you can assign the codelist with the respective Code with their value and Name which can also be nested, in this case the system will automatically assign them the ParentCode

## Frequency

The dimension of type frequency is required for the system OnTheFly since all data requests are type TimeSeries. But it is still possible that in a given dataflow this dimension is not present. In this case the system will automatically add the Frequency Dimension and the respective codelist picking her recover from the file that is located in

**“Bin/ConfigurationXml/FrequencyCodelist.xml”**

# **Log Management**

In OnTheFly system is a part of the configuration dedicated to the log

|  |  |
| --- | --- |
| LogLocation | Determinate a location of application Log. Is possible use a special Word as %Temp% for identify a temporary Path |
| LogLevel | Possible value:  **"None"** -> No log  **"All"** -> All log (Debug Mode)  **"Warning**" -> Only Warning and Error Log  **"Error"** -> Only Error Log |

If the configuration is properly configured: LogLevel not None and LogLocation is an existing folder and not ReadOnly for the service, the system will write a log file every hour (if operations are performed).

Will be provided by the user consultation and the cancellation of the same