TinyRegistry
Repository Service Guide

V 1.0.1

0 Table

1 Introduction

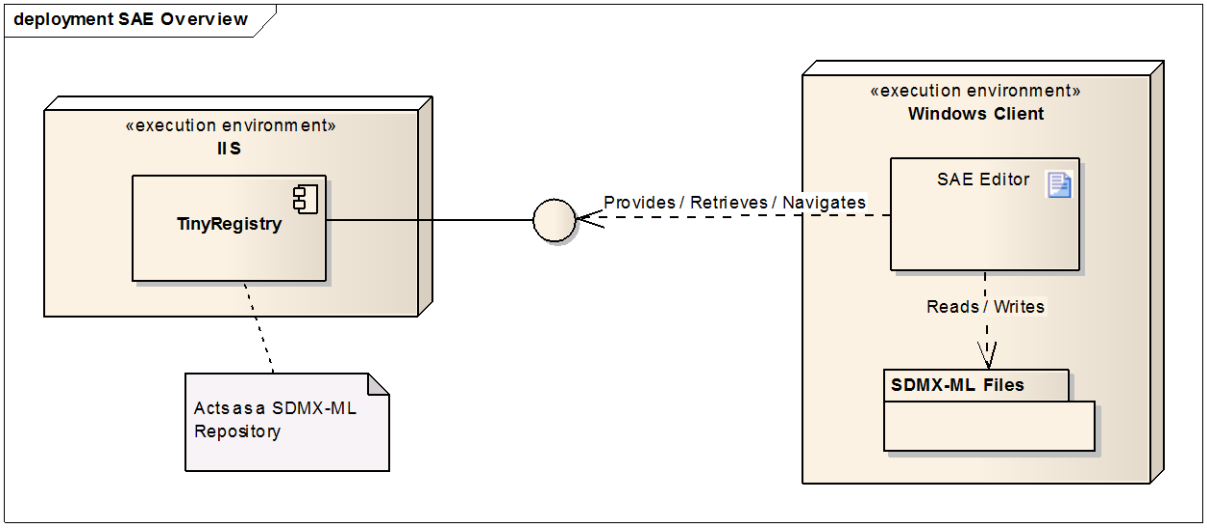
This document serves as a technical specification of the repository component of the SAE SDMX-ML Editor system. The repository is a .NET 2.0 web service of the TinyRegistry. It must be deployed under Internet Services **IIS 5.1 or later**. It requires Microsoft **.NET runtime 2.0 or later**.
This service provides all operations required by the SAE Editor in order to manage the metadata artefacts in a centralized manner. It is also a distribution point for the SDMX-ML artefacts.

1.1 RESTful service

A RESTful service is also provided. It acts as a wrapper on the TinyRegistry. It allows consulting SDMX-ML documents according to the syntax defined in the *SDMX Guidelines for web services , April 2011 version 2.1.*
More detail in paragraph .RESTful Web Service description

2 System Context

The Application Server (IIS) contains the TinyRegistry repository service and all artefact documents managed through the web service.



The repository manages structured artefacts by artefact type. It supports internationalization and must be configured according to the SAE editor internationalisation subset support. A set of languages must be specified when the repository is empty and cannot be changed once it is used.

All artefacts are organized by artefact Type in file structure under a root directory which is specified in the configuration file. **Write & create permissions for the web service access should be specified for the root directory which contains all artefact documents.** This directory is also specified in the web.config file of the repository.

2.1 Demonstration passwords

For this www.sdmx.ch distribution, the operations password is always fixed at: `44533231702B32`

SAE user Password is now removed	

2.2 TinyRegistry configuration file

The XML file named web.config is used to parametrize the TinyRegistry Services.

Location	Tag	Description
<configuration> <appSettings>	<add key="ArtefactRootPath" value=" SAMPLE value: value="C:\SAE_SRC\ Artefacts_full"	The directory root path under which all types of managed artefacts will be located. A sub directory exists for each type of artefact. See the samples directories: *Artefacts_full* or *artefacts_empty*
<configuration> <appSettings>	<add key="SupportedLanguages" value= SAMPLE value: value="en;de;fr;it"	Defines the set of supported languages. It must match the set defined in SAE Editor.
<configuration> <appSettings>	<add key="GroupOrganisationScheme" value= SAMPLE value: value="NSI+ORG_NSI+1.0"	Optional. Is filled with the NSI organization scheme including NSI owners groups for managing metadata ownership. The value must match the Full Id of the organisationScheme.
<configuration> <appSettings>	<add key="GroupFullAdministrators" value=" SAMPLE value: value="U4578;U0021"	Optional Must be filled for using Ownership support. It consists of the list of user ID separated by a semi column. These will be the repository administrators.
<configuration> <appSettings>	<add key="AutomaticBackup" value="False" />	Indicates if backup on already submitted artifact is done automatically or on user demand only. Default is on demand (false)
<configuration> <appSettings>	<add key="SlidingBackupSize" value="5" />	The backup is incremental on each artifact. This parameter indicates the number of backup that are conserved in sliding history.

3 Web Service description

This chapter describes the SOAP WebService offered by the Tiny Registry system in details and defines its functionality, its parameters, etc.

Note: The Service URI given in the following documentation is based on an IT implementation. It can differ according to your specific publication of the service.

Currently the following Service Operations are being offered:

- BackupAutomaticEnable
- CheckOut
- DeleteArtefact
- DirInfo
- ExistArtefact
- GetArtefact
- GetArtefactBackup
- GetArtefactBackupList
- GetArtefactOS
- IsAdministrator
- SetDisseminationStatus
- SubmitArtefact
- SubmitArtefactBackup
- Test
- UndoCheckOut

3.1 Repository

3.1.1 Distinct Service Identification

Service Name	TinyRegistry/Repository
Service Version	1.0.0
Service functionality	This service provides all the operations required to manage the repository of SDMX-ML artefacts documents.
Service functionality Version	1.0.0
Service URI	To be defined in your NSI. Sample: http://SAE/MyNSI/TinyRegistry/Repository.asmx

3.1.2 Detailed functionality description

The Repository service manages SDMX-ML artefacts. Each artefact is identified by its full id. That is: Agency ID + Artefact ID + Artefact Version. All artefacts are managed into the predefined root directory as specified in the web config of the service. Each type of artefact (e.g. Codelists, ConceptSchemes, MSD, etc.) is located under a sub-directory named after the artefact type.

Generally, an SDMX-ML document contains only the identified artefact. However, it is also possible to manage aggregated SDMX-ML artefacts in packages. In such case, the set of artefacts present in the document can only be retrieved as a package identified by the artefact's full ID, which is defined (selected) during the first submission of the agregated SDMX-ML document.

The artefact types supported by the service are identified by the following strings:

OrganisationSchemes,
Dataflows,
Metadataflows,
CategorySchemes,
CodeLists,
HierarchicalCodelists,
Concepts,
MetadataStructureDefinitions,
KeyFamilies,
StructureSets,
ReportingTaxonomies,
Processes

3.1.2.1 Maintained informations

The service maintains a set of information about each artefact:

INFO	Comment	Sample
ID	Artefact ID	CL_FREQ
AI	Artefact Agency ID	SDMX
VER	Artefact Version	1.0
VF	Artefact valid from	2010-01-01
VT	Artefact valid to	2010-12-31
Name_xx	Artefact name in each supported language (one tag per language). The set of supported	Code list for Frequency

	languages is specified in the service web configuration file.	
IF	Artefact is Final or not	true
LU	Last update in repository (YYYY-MM-DD)	2010-07-01
UB	Last update By (Windows Identity Name)	mediaRoom/prati
CO	Check out (Reserved) by (Windows Identity Name)	EDI/U67878
CA	Check out in repository at (YYYY-MM-DD)	2010-07-12
DI	(informative, the default is Intern) Diffusion Status available codes: 0 – No 1 – Public 2 - Intern	2

3.1.3 Optional Ownership management

The service can optionally manage the artefacts ownership inside your NSI organization. In this case you must provide an OrganisationScheme document to the service. This organization scheme must contain a hierarchical structure of Agencies where the leaves are the different groups owning some NSI artefacts. Inside each group the Maintenance contact is required. Other collaborators can also be added into the group as other persons for Contact. Each person defined as Contact must have his ID filled in together with his MS Window identity name. For example for a Window identity of 'EDI/U0021' the ID is filled with: U0021. Once defined and used, the NSI OrganisationScheme could still be modified by its owners. However, such modifications should be done with caution, i.e the existing groups which already posses SDMX artefacts should not be removed from the Agencies hierarchy or their IDs modified.

It is also required to define one or more administrators for the file repository. The administrators are allowed to operate on any artefact, such that they could mainatine for example non NSI artefacts (the SDMX-ML artefacts provided by external to the NSI organisation institutions). In order to activate the ownership management, the service configuration file must contain values in the following keys:

```
<!-- Support for Ownership managment -->
<add key="GroupOrganisationScheme" value="" />          for ex: value="NSI+ORG_NSI+1.0
<add key="GroupFullAdministrators" value="" />          for ex: value="U4578; U3422"
```

The Ownership support is disabled when these values are empty (or as long as the GroupOrganisationScheme is not submitted to the repository).

Requirements for ownership support

Request	Description
1.	The support for management of Group Ownership can be activated in the configuration file of the TinyRegistry services.
2.	The Repository administrators can manage every metadata of the Repository.
3.	The Repository administrators are defined in the configuration file of the TinyRegistry services.
4.	It is required to have, at least, one repository administrator.
5.	Only Repository administrator can manage non NSI artefacts.
6.	The Full ID of the Organization Scheme where Ownership Agencies, Groups and Users are defined is specified in the configuration file of TinyRegistry services

7.	As long as this Organization Scheme is not present in the repository, the service acts as if group ownership support is deactivated.
8.	When the content of this organization scheme is updated, the service takes care of the modifications.
9.	NSI Metadata owner Groups are defined in the OrganisationScheme – Agency as leaves of the Agency Hierarchy.
10.	NSI Metadata owners are defined in the Organisation – Agency – Group - Contacts area of the Organization Scheme.
11.	The Group administrator of the group is defined as the Maintenance contact of the group. It is required for defining an owner group. Others members of the group are defined in the Contact – Others area.
12.	Each person of the group must have its ID matching its MS Windows user identity. For example ***mediaRoom\prati*** has the ID defined as: **prati**
13.	Only the Group administrator (maintenance contact) is allowed to change the diffusion status of an artefact owned by the group.
14.	When Group Ownership support is activated, the web service method "*GetArtefact*" will not provides artefacts with a "no diffusion" status set.
15.	A web service method "*GetArtefactOS*" is provided to allow owners of a non diffused artefact to access it in reading mode.
16.	From 1.0 An incremental backup mechanism is provided for each artifact.

3.1.4 CheckOut Service operation functionality description

CheckOut is part of the lock/unlock mechanism available for modifying artefacts already present in the registry. It is provided in order to avoid that different users operate on the same artefact in same time. CheckOut is the operation of reservation of an artefact by a specific Windows user.

The unlock is automatically done when the user submit the updated artefact document, or if the user invoke the ''UndoCheckOut' operation

Name of service operation	Checkout
Version (optional)	
Grouping (optional)	
Behavior	Reserves (lock) the specified artefact for the user with indicated Windows Identity. Only this user can then modifiy or unlock the specified artefact.
Preconditions	The artefact exists in repository The artefact is not already reserved by some other user. The user is part of the agency group that owns the artefact (only in case of ownership support)
Postconditions	The artefact is reserved for specified Window identity. The service keeps track of date and identity of reservation.
Errors	General access error Invalid artefact type Artefact does not exist Artefact already reserved. Unauthorized user according to artefact agencyId (in case of ownership support)

3.1.4.1 CheckOut Interface

- Request / Responces, synchronous

Request Message

Parameter name	Data type	Card	Description / semantics
user	string		Windows Identity
pwd	string		Valid Password according to authorization mechanism implemented in service.
artType	string		A string matching one of the supported Artefact types.
fullId	string		AgencyID + Artefact ID + Artefact Version Ex: “ESTAT+CL_FREQ+1.0”

Response Message

Parameter name	Data type	Card	Description / semantics
StatusMessage	XML Element		The XmlElement contains a Sucess status code. <TinyRegistry> <StatusMessage status="**Success**" /> </TinyRegistry>

Error Message

Parameter name	Data type	Card	Description / semantics
StatusMessage	XML Element		The XmlElement contains a failure status code and an explanatory message. <TinyRegistry> <StatusMessage status="**Failure**"> <MessageText>**Unknown Artefact ESTAT+CL_FREQ+1.1**</MessageText> </StatusMessage> </TinyRegistry>

3.1.5 UndoCheckOut Service operation functionality description

This operation cancels any previously reservation done on the specified artefact. It is not an error to operate on a non reserved artefact.

Name of service operation	UndoCheckout
Version (optional)	
Grouping (optional)	
Behavior	Cancel Reservation by user on specified artefact.
Preconditions	The artefact exists in repository

	The artefact was reserved by same user. The user is part of the agency group that owns the artefact (only in case of ownership support)
Postconditions	The artefact is no more reserved. The service keeps track of cancelled reservation
Errors	General access error Invalid artefact type Artefact does not exist Unauthorized user according to artefact agencyId (in case of ownership support)

3.1.5.1 UndoCheckOut Interface

- Request / Responces, synchronous

Request Message

Parameter name	Data type	Card	Description / semantics
user	string		Windows Identity
pwd	string		Valid Password according to authorization mechanism implemented in the service.
artType	string		A string matching one of the supported Artefact types.
fullId	string		AgencyID + Artefact ID + Artefact Version Ex: “ESTAT+CL_FREQ+1.0”

Response Message

Parameter name	Data type	Card	Description / semantics
StatusMessage	XML Element		The XmlElement contains a Sucess status code. <TinyRegistry> <StatusMessage status="**Success**" /> </TinyRegistry>

Error Message

Parameter name	Data type	Card	Description / semantics
StatusMessage	XML Element		The XmlElement contains a failure status code and an explanatory message. <TinyRegistry> <StatusMessage status="**Failure**"> <MessageText>**Unknown Artefact ESTAT+CL_FREQ+1.1**</MessageText> </StatusMessage> </TinyRegistry>

3.1.6 ExistArtefact Service operation functionality description

This operation allows interrogating repository about existance of an artefact of a specified type.

Name of service operation	ExistArtefact
Version (optional)	
Grouping (optional)	
Behavior	Returns a Success status if an artefact of specified type holds the given full Id parameter
Preconditions	The artefact exists in repository
Postconditions	Informativ, no changes in Repository
Errors Warning	General access error Invalid artefact type Artefact does not exist (As Warning)

3.1.6.1 ExistArtefact Interface

- Request / Responces, synchronous

Request Message

Parameter name	Data type	Card	Description / semantics
artType	string		A string matching one of the supported Artefact types.
fullId	string		AgencyID + Artefact ID + Artefact Version Ex: “ESTAT+CL_FREQ+1.0”

Response Message

Parameter name	Data type	Card	Description / semantics
StatusMessage	XML Element		The XmlElement contains a Sucess status code. <TinyRegistry> <StatusMessage status="**Success**" /> </TinyRegistry>

Warning Message

Parameter name	Data type	Card	Description / semantics
StatusMessage	XML Element		The XmlElement contains a warning status code and an explanatory message. <TinyRegistry> <StatusMessage status="**Warning**"> <MessageText>**Unknown Artefact: codelists, ESTAT+CL_FREQ+1.1** </MessageText> </StatusMessage> </TinyRegistry>

Error Message

Parameter name	Data type	Card	Description / semantics
StatusMessage	XML Element		The XmlElement contains a failure status code and an explanatory message. <TinyRegistry> <StatusMessage status="**Failure**"> <MessageText>**Invalid Artefact Type: Survey**</MessageText> </StatusMessage> </TinyRegistry>

3.1.7 DirInfo Service operation functionality description

This operation provides the list of managed artefacts of the specified type. The list contains all the items of the maintained information. This list is embedded as an xml element representing a .NET DataSet with its schema.

Name of service operation	DirInfo
Version (optional)	
Grouping (optional)	
Behavior	Returns an Information list of all artefacts of specified type. The list is an XML element with embedded scheme
Preconditions	The artefact type is supported by service
Postconditions	Informativ, no changes in Repository
Errors	General access error Invalid artefact type as empty Dataset

3.1.7.1 DirInfo Interface

- Request / Responces, synchronous

Request Message

Parameter name	Data type	Card	Description / semantics
artType	string		A string matching one of the supported Artefact types.

Response Message

Parameter name	Data type	Card	Description / semantics
Dataset	Dataset as xml element with embedded schema		The dataset representing the list with an element matching the specified artefact type. - <DataSet xmlns="**http://fmr/sms/sae.ch/**"> - <xs:schema id="**NewDataSet**" xmlns="" xmlns:xs="**http://www.w3.org/2001/XMLSchema**" xmlns:msdata="**urn:schemas-microsoft-com:xml-msdata**"> - <xs:element name="**NewDataSet**" msdata:IsDataSet="**true**"

			msdata:UseCurrentLocale="**true**"> - <xs:complexType> - <xs:choice minOccurs="**0**" maxOccurs="**unbounded**"> - <xs:element name="**CodeLists**"> . . .

Error Message

Parameter name	Data type	Card	Description / semantics
Dataset	XML Element		The XmlElement contains an empty Dataset

3.1.8 GetArtefact Service operation functionality description

This operation returns, in a SDMX Message document element, the artefact type instance matching the specified full Id. In case of error, an invalid status message is returned.

Caution: In case of Ownership support, this method does not allow to retrieve artefact with a NO diffusion status.'This operation is not used by associated SAE editor, instead it uses the 'GetArtefactOS' operation.

.

Name of service operation	GetArtefact
Version (optional)	
Grouping (optional)	
Behavior	Returns a SDMX Message document element containing the specified artefact information
Preconditions	The artefact type is supported The artefact exists in repository The artefact does not have a 'NO Diffusion' status (only in case of ownership support)
Postconditions	Informativ, no changes in Repository
Errors	General access error Invalid artefact type Artefact does not exist

3.1.8.1 GetArtefact Interface

- Request / Responces, synchronous

Request Message

Parameter name	Data type	Card	Description / semantics
artType	string		A string matching one of the supported Artefact types.
fullId	string		AgencyID + Artefact ID + Artefact Version Ex: "ESTAT+CL_FREQ+1.0"

Response Message

Parameter name	Data type	Card	Description / semantics
SDMX-ML	SDMX-ML document as XML Element		The SDMX ML containing requested artefact. The document contains others artefacts in case of aggregated document.

Error Message

Parameter name	Data type	Card	Description / semantics
StatusMessage	XML Element		The XmlElement contains a failure status code and an explanatory message. <TinyRegistry> <StatusMessage status="**Failure**"> <MessageText>**Unknown Artefact ESTAT+CL_FREQ+1.1**</MessageText> </StatusMessage> </TinyRegistry>

3.1.9 GetArtefactBackup Service operation functionality description

This operation returns, in a SDMX Message document element, the required backup of the artefact type instance matching the specified full Id. In case of error, an invalid status message is returned.

.

Name of service operation	GetArtefact
Version (optional)	
Grouping (optional)	
Behavior	Returns a SDMX Message document element containing the specified backup version of artefact information
Preconditions	The artefact type is supported The artefact exists in repository The specified artefact backup exists
Postconditions	Informativ, no changes in Repository
Errors	General access error Invalid artefact type Artefact does not exist

3.1.9.1 GetArtefactBackup Interface

- Request / Responces, synchronous

Request Message

Parameter name	Data type	Card	Description / semantics
user	string		Windows Identity
pwd	string		Valid Password according to authorization mechanism implemented in service.
artType	string		A string matching one of the supported Artefact types
fullId	string		AgencyID + Artefact ID + Artefact Version

			Ex: “ESTAT+CL_FREQ+1.0”
VersionNumber			Incremental backup number desired

Response Message

Parameter name	Data type	Card	Description / semantics
SDMX-ML	SDMX-ML document as XML Element		The SDMX ML containing requested artifact in its backup state. The document contains others artefacts in case of aggregated document.

Error Message

Parameter name	Data type	Card	Description / semantics
StatusMessage	XML Element		The XmlElement contains a failure status code and an explanatory message. <TinyRegistry> <StatusMessage status="**Failure**"> <MessageText>**Unknown Artefact ESTAT+CL_FREQ+1.1**</MessageText> </StatusMessage> </TinyRegistry>

3.1.10 GetArtefactBackupList Service operation functionality description

This operation retieves the list of all backups for an Artefact. This list is embedded as an xml element representing a .NET DataSet with its schema.

Name of service operation	GetArtefactBackupList
Version (optional)	1.0
Grouping (optional)	
Behavior	Returns an Information list of all backups available for an artefact of specified type. The list is an XML element with embedded scheme
Preconditions	The artefact type is supported by service
Post conditions	Informative, no changes in Repository
Errors	General access error Empty list as empty Dataset

3.1.10.1 GetArtefactBackupList Interface

- Request / Responses, synchronous

Request Message

Parameter name	Data type	Card	Description / semantics
artType	string		A string matching one of the supported Artefact types.
fullId	string		AgencyID + Artefact ID + Artefact Version Ex: “ESTAT+CL_FREQ+1.0”

Response Message

Parameter name	Data type	Card	Description / semantics
Dataset	Dataset as xml element with embedded schema		The dataset representing the of backup matching the specified artefact type. . . .

Error Message

Parameter name	Data type	Card	Description / semantics
Dataset	XML Element		The XmlElement contains a nil Dataset

3.1.11 GetArtefactOS Service operation functionality description

This operation returns, in a SDMX Message document element, the artefact type instance matching the specified full Id. In case of error, an invalid status message is returned.

In case of Ownership support, this method allows to retrieve artefact with 'NO diffusion' status.The retrieval is allowed for users that are members of the agency group of the artefact as well as for repository administrator(s).

This operation is used by SAE editor in replacement of the operation 'GetArtefact'.

The SAE editor, when retrieving an artefact, checks the references to others artefacts contained in the artefact and makes the subsequent calls to load the referred information.

.

Name of service operation	GetArtefact
Version (optional)	
Grouping (optional)	
Behavior	Returns a SDMX Message document element containing the specified artefact information
Preconditions	The artefact type is supported The artefact exists in repository If the artefact has 'NO Diffusion' status the user must be a repositora administrator or a member of the artefact agency group (only in case of ownership support)
Postconditions	Informativ, no changes in Repository
Errors	General access error Invalid artefact type Artefact does not exist Unauthorized user according to artefact agencyId (in case of ownership support)

3.1.11.1 GetArtefactOS Interface

- Request / Responces, synchronous

Request Message

Parameter name	Data type	Card	Description / semantics
user	string		Windows Identity
pwd	string		Valid Password according to authorization mechanism implemented in service.
artType	string		A string matching one of the supported Artefact types
fullId	string		AgencyID + Artefact ID + Artefact Version Ex: “ESTAT+CL_FREQ+1.0”

Response Message

Parameter name	Data type	Card	Description / semantics
SDMX-ML	SDMX-ML document as XML Element		The SDMX ML containing requested artefact. The document contains others artefacts in case of aggregated document.

Error Message

Parameter name	Data type	Card	Description / semantics
StatusMessage	XML Element		The XmlElement contains a failure status code and an explanatory message. <TinyRegistry> <StatusMessage status="**Failure**"> <MessageText>**Unknown Artefact ESTAT+CL_FREQ+1.1** </MessageText> </StatusMessage> </TinyRegistry>

3.1.12 SubmitArtefact Service operation functionality description

This operation stores a new SDMX-ML document containing the artefact specified in the request parameters. If the artefact is already present in the repository, it must be reserved by the user before it can be submitted for an update.

When ownership support is activ, only users members of the agency group of the artefact can submit update of the artefact. It is always possible for repositoty administrator to submit artefacts.

The SDMX-ML document must contain the artefact type and Identification that is given in request parameters. It is allowed to submit document that contains others artefacts, but they can only be retrieved as a whole through the id used in request parameter.

.

Name of service operation	SubmitArtefact
Version (optional)	
Grouping (optional)	
Behavior	Submits a new artefact to registry or updates existing one (if checked out by same user)
Preconditions	The artefact type is supported The artefact exists in the submitted SDMX-ML document

	In case of update, the artefact is already checked out In case of update, the user is member of the agency group of the artefact
Postconditions	Informativ, no changes in Repository
Errors	General access error Invalid artefact type Artefact is not reserved (checked out) Unauthorized user according to artefact agencyId (in case of ownership support)

3.1.12.1 SubmitArtefact Interface

- Request / Responces, synchronous

Request Message

Parameter name	Data type	Card	Description / semantics
user	string		Windows Identity
pwd	string		Valid Password according to authorization mechanism implemented in service.
artType	string		A string matching one of the supported Artefact types.
fullId	string		AgencyID + Artefact ID + Artefact Version Ex: “ESTAT+CL_FREQ+1.0”
elArtefact	xml		XML element representing a valid SDMX-ML document

Response Message

Parameter name	Data type	Card	Description / semantics
StatusMessage	XML Element		The XmlElement contains a Sucess status code. <TinyRegistry> <StatusMessage status="**Success**" /> </TinyRegistry>

Error Message

Parameter name	Data type	Card	Description / semantics
StatusMessage	XML Element		The XmlElement contains a failure status code and an explanatory message. <TinyRegistry> <StatusMessage status="**Failure**"> <MessageText> **Unable to locate Artefact NSI+CL_QUALITY+1.0 in submitted XML Document.** </MessageText> </StatusMessage> </TinyRegistry>

3.1.13 SubmitArtefactBackup Service operation functionality description

This operation stores a new SDMX-ML document containing the artefact specified in the request parameters. If the artefact is already present in the repository it is copied as a backup file with incremental number before the submitted artifact. Note that artefact must be reserved by the user before it can be submitted for an update.

When ownership support is activ, only users members of the agency group of the artefact can submit update of the artefact. It is always possible for repositoty administrator to submit artefacts.

The SDMX-ML document must contain the artefact type and Identification that is given in request parameters. It is allowed to submit document that contains others artefacts, but they can only be retrieved as a whole through the id used in request parameter.

.

Name of service operation	SubmitArtefact
Version (optional)	
Grouping (optional)	
Behavior	Submits a new artefact to registry or updates existing one (if checked out by same user)
Preconditions	The artefact type is supported The artefact exists in the submitted SDMX-ML document In case of update, the artefact is already checked out In case of update, the user is member of the agency group of the artefact
Postconditions	Informativ, no changes in Repository A backup is created with previous document content.
Errors	General access error Invalid artefact type Artefact is not reserved (checked out) Unauthorized user according to artefact agencyId (in case of ownership support)

3.1.13.1 SubmitArtefactBackup Interface

- Request / Responces, synchronous

Request Message

Parameter name	Data type	Card	Description / semantics
user	string		Windows Identity
pwd	string		Valid Password according to authorization mechanism implemented in service.
artType	string		A string matching one of the supported Artefact types.
fullId	string		AgencyID + Artefact ID + Artefact Version Ex: “ESTAT+CL_FREQ+1.0”
elArtefact	xml		XML element representing a valid SDMX-ML document

Response Message

Parameter name	Data type	Card	Description / semantics
StatusMessage	XML Element		The XmlElement contains a Sucess status

			code. <TinyRegistry> <StatusMessage status="**Success**" /> </TinyRegistry>

Error Message

Parameter name	Data type	Card	Description / semantics
StatusMessage	XML Element		The XmlElement contains a failure status code and an explanatory message. <TinyRegistry> <StatusMessage status="**Failure**"> <MessageText> **Unable to locate Artefact NSI+CL_QUALITY+1.0 in submitted XML Document.** </MessageText> </StatusMessage> </TinyRegistry>

3.1.14 DeleteArtefact Service operation functionality description

This operation allows removing the specified artefact from the repository.

Caution: This will not remove artefacts referred by the targeted artefact. It is the responsibility of the user to decide if referenced artefacts must also be removed.

When Ownership is managed by the service, only members of the artefact agency group or repository administrator(s) can remove the artefact.

.

Name of service operation	DeleteArtefact
Version (optional)	
Grouping (optional)	
Behavior	Deletes selected artefact, event if currently checked-out.
Preconditions	The artefact type is supported The artefact exists in repository In case of ownership management, the user is the maintenance contact of the agency group.
Postconditions	The status associated to the artefact is updated
Errors	General access error Invalid artefact type Artefact is not part of repository Unauthorized user according to artefact agencyId (in case of ownership support)

3.1.14.1 DeleteArtefact Interface

- Request / Responces, synchronous

Request Message

Parameter name	Data type	Card	Description / semantics
user	string		Windows Identity
pwd	string		Valid Password according to authorization mechanism implemented in service.
artType	string		A string matching one of the supported Artefact type
fullId	string		AgencyID + Artefact ID + Artefact Version Ex: “ESTAT+CL_FREQ+1.0”
diffusionCd	string		A one character string representing the number coding the status 0 – No 1- Public 2 - Internal

Response Message

Parameter name	Data type	Card	Description / semantics
StatusMessage	XML Element		The XmlElement contains a Sucess status code. <TinyRegistry> <StatusMessage status="**Success**" /> </TinyRegistry>

Error Message

Parameter name	Data type	Card	Description / semantics
StatusMessage	XML Element		The XmlElement contains a failure status code and an explanatory message. <TinyRegistry> <StatusMessage status="**Failure**"> <MessageText> **Unable to locate Artefact NSI+CL_QUALITY+1.0 in submitted XML Document.** </MessageText> </StatusMessage> </TinyRegistry>

3.1.15 SetDisseminationStatus Service operation functionality description

This operation allows updating the dissemination status associated to artefacts managed in repository. By default an artefact that is submitted in repository holds the “Internal” status.

With this operation it is possible to modify the status. Availables statuses are:

0 – No diffusion

1 – Public

2 - Internal

When Ownership is managed by the service, artefact with “NO diffusion” can only be accessed by the users of the agency group that owns the artefact (or by repository administrators).

In this case it is also only the Maintenance contact of the group that is allowed to modify the status associated to the artefact (or the repository administrator(s))

.

Name of service operation	SetDisseminationSrtatus
Version (optional)	
Grouping (optional)	
Behavior	Defines Dissemination status for artefact.(0-NO, 1-Public, 2-Intern)
Preconditions	The artefact type is supported The artefact exists in repository The status code is valid In case of ownership management, the user is the maintenance contact of the agency group.
Postconditions	The status associated to the artefact is updated
Errors	General access error Invalid artefact type Invalid status code Artefact is not part of repository Unauthorized user according to artefact agencyId (in case of ownership support)

3.1.15.1 SetDisseminationStatus Interface

- Request / Responces, synchronous

Request Message

Parameter name	Data type	Card	Description / semantics
user	string		Windows Identity
pwd	string		Valid Password according to authorization mechanism implemented in service.
artType	string		A string matching one of the supported Artefact types
fullId	string		AgencyID + Artefact ID + Artefact Version Ex: “ESTAT+CL_FREQ+1.0”
diffusionCd	string		A one character string representing the number coding the status 0 – No 1- Public 2 - Internal

Response Message

Parameter name	Data type	Card	Description / semantics
StatusMessage	XML Element		The XmlElement contains a Sucess status code.

			<TinyRegistry> <StatusMessage status="**Success**" /> </TinyRegistry>

Error Message

Parameter name	Data type	Card	Description / semantics
StatusMessage	XML Element		The XmlElement contains a failure status code and an explanatory message. <TinyRegistry> <StatusMessage status="**Failure**"> <MessageText>**Invalid diffusion status code: 4** </MessageText> </StatusMessage> </TinyRegistry>

3.1.16 Test Service operation functionality description

This operation allows retrieving the version of the Service.

.

Name of service operation	Test
Version (optional)	
Grouping (optional)	
Behavior	Provides service version.
Preconditions	
Postconditions	The version is returned
Errors	General access error

3.1.16.1 Test Interface

- Request / Responces, synchronous

Request Message

Parameter name	Data type	Card	Description / semantics

Response Message

Parameter name	Data type	Card	Description / semantics
StatusMessage	XML Element		The XmlElement contains a Sucess status code. <TinyRegistry> <StatusMessage status="**Success**"> <MessageText>**TinyRegistry version: 0.9.4.24911** </MessageText> </StatusMessage>

			</TinyRegistry>

4 Quick evaluation with SAE Editor

This chapted describes how to proceed to a quick overview and evaluation of the solution linked to the SAE editor. This evaluation allows you to skip the TinyRegistry installation to an IIS application server. It allows a simulation of the service on the same machine that holds the SAE Editor.
Of course, for a real usage, the service has to be deployed in IIS application server.

4.1 Copy SAE Install root dir on your local drive

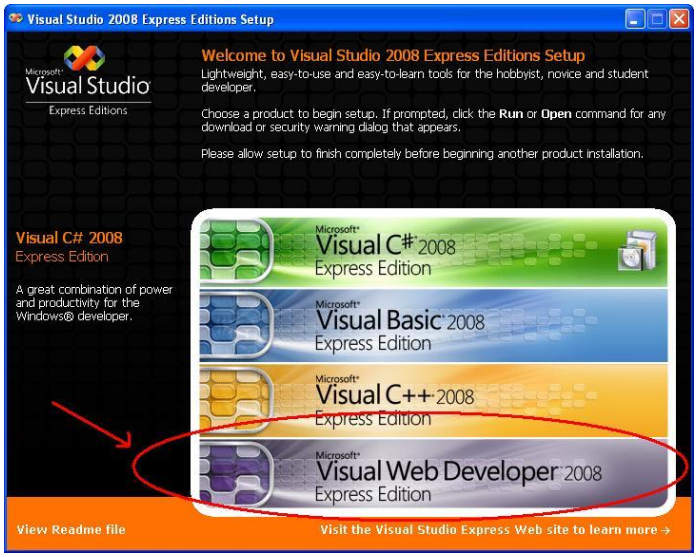
If you intend to use only SAE editor without Repository support, you can ignore most of this documentation. See only: 1.4 SAE Editor configuration

4.2 Install .NET runtime

4.2.1 Install Microsoft visual web developer express

This will allows running *TinyRegistry* for quick trial without an IIS installation.
Note: For a real usage, the TinyRegistry must be deployed under IIS server environment.

Get free Visual studio 2008 from: http://www.microsoft.com/express/downloads/
Run installation.

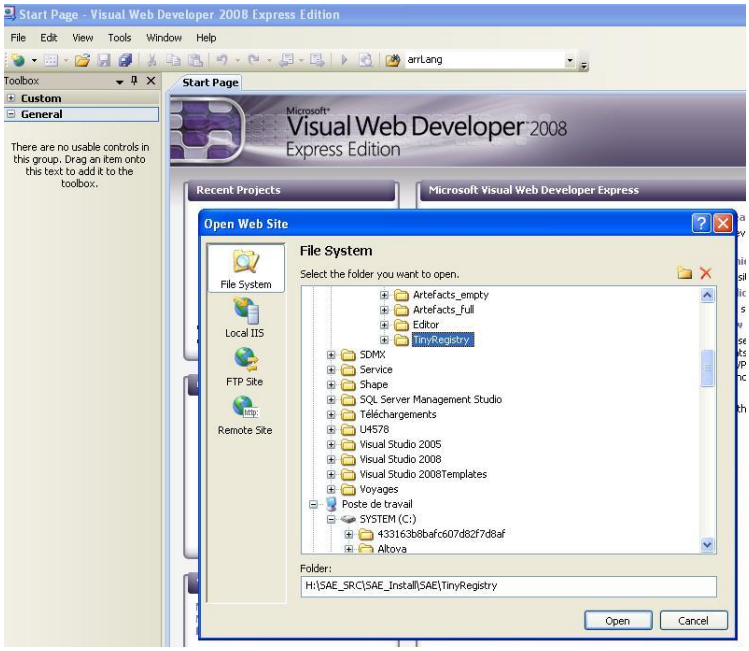


Select **Visual Web Developer**

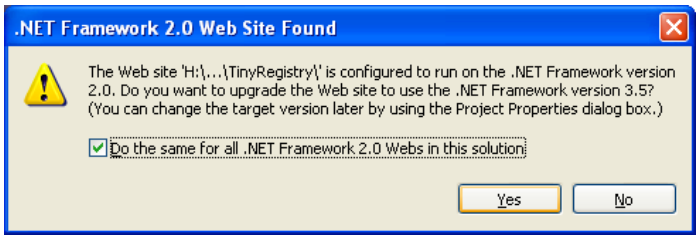
4.3 Configure TinyRegistry

Once dot Net and MS Developer are fully installed run **Visual Web Developer.**

With the command: Menu File – Open Web Site, select the TinyRegistry folder



The application signals that the site is configured for .NET 2.0 and asks for updating.



Answers: **No**

Selects **web.config** in the solution Explorer

Change the "**SupportedLanguages**" value if needed.

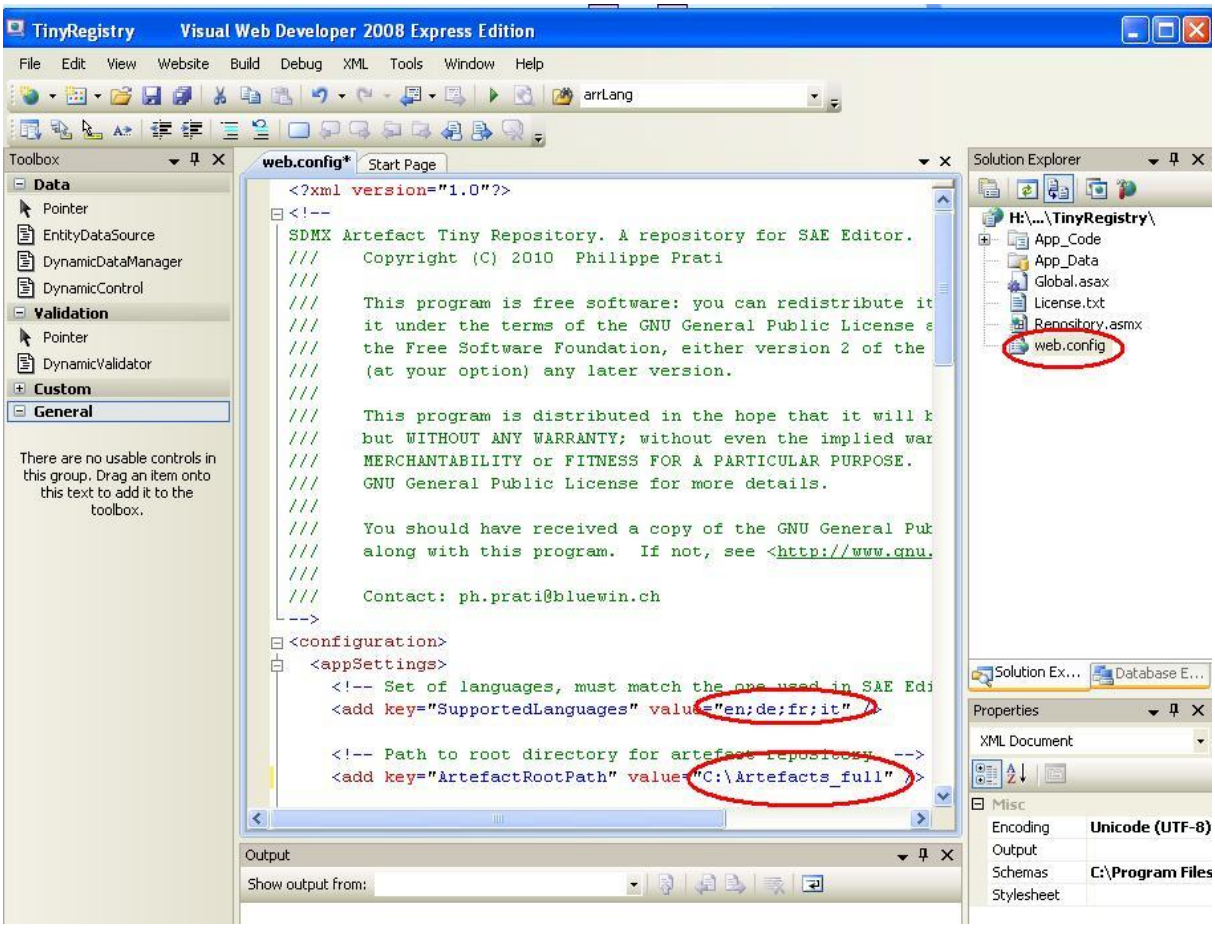
Change the "**ArtefactRootPath**" value to reflect the root directory for artefacts.

Select **Artefacts_empty** folder to start from scratch

```
<!-- Path to root directory for artefact repository. -->
    <add key="ArtefactRootPath" value="C:\SAE_SRC\SAE_Install\SAE\Artefacts_empty" />
```

- Or selects **Artefacts_full** to uses sample files already registered in repository.

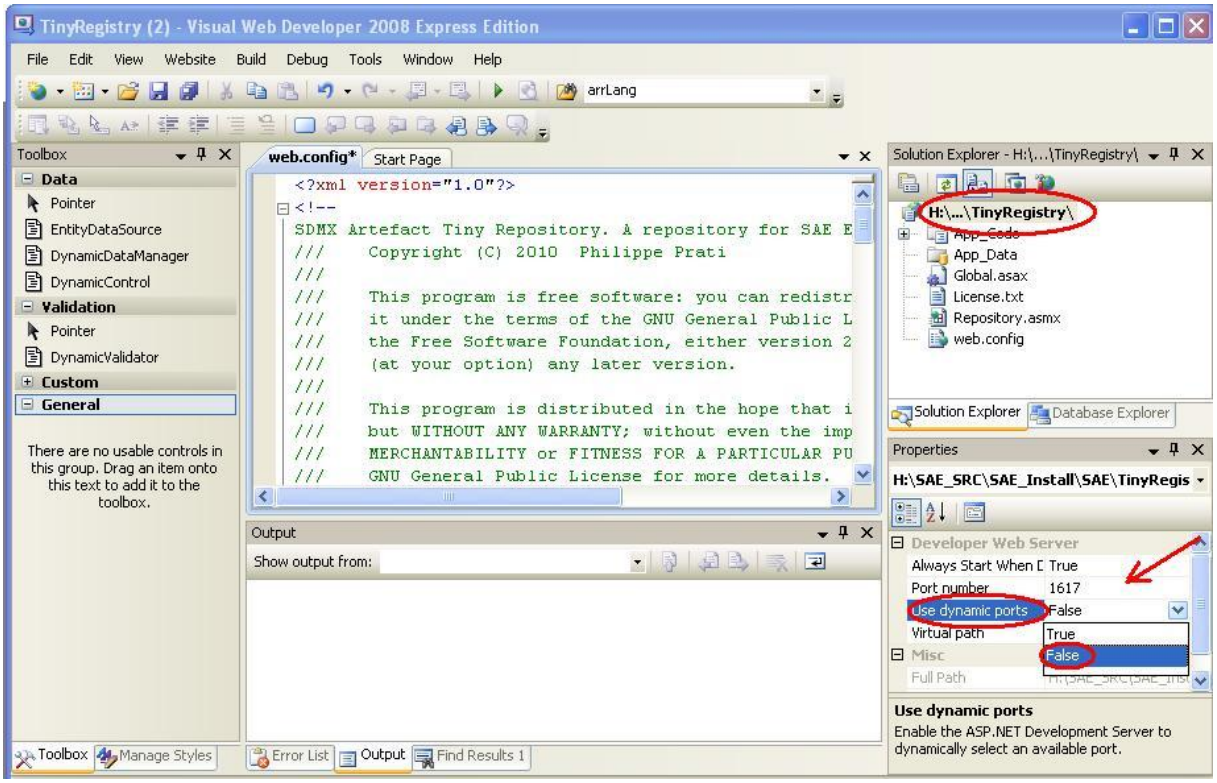
```
<!-- Path to root directory for artefact repository. -->
<add key="ArtefactRootPath" value="C:\SAE_SRC\SAE_Install\SAE\Artefacts_full" />
```



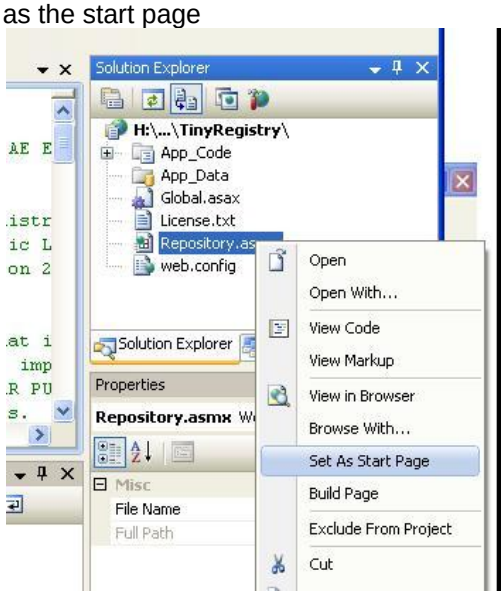
Now indicates to the web service to uses always the same port number:

Selects **TinyRegistry** in the solution Explorer

Change the "**Use dynamic ports**" value to **false**.

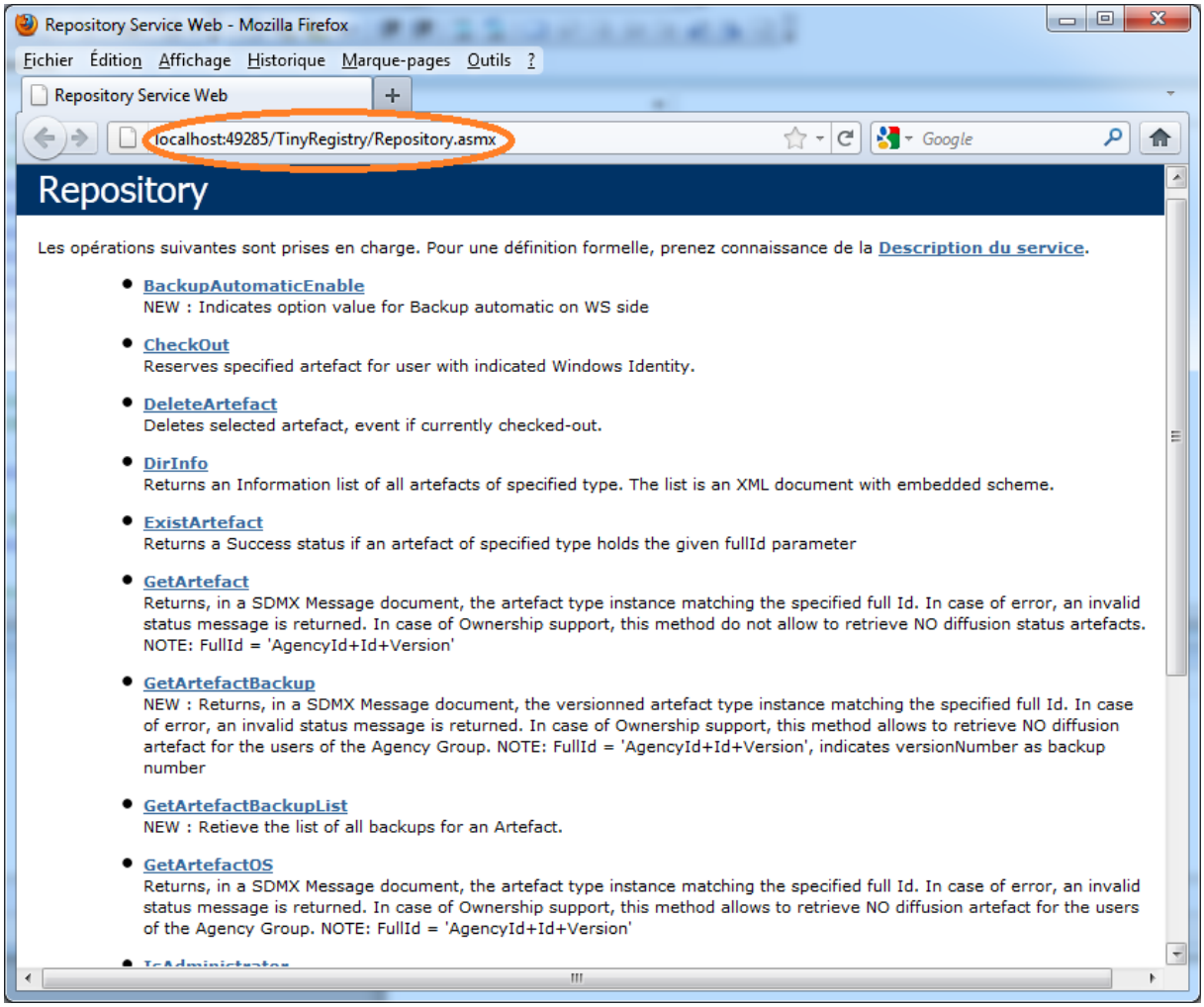


Indicates **Repository.asmx** as the start page



You can now **run** the Service:

The following page should be displayed:

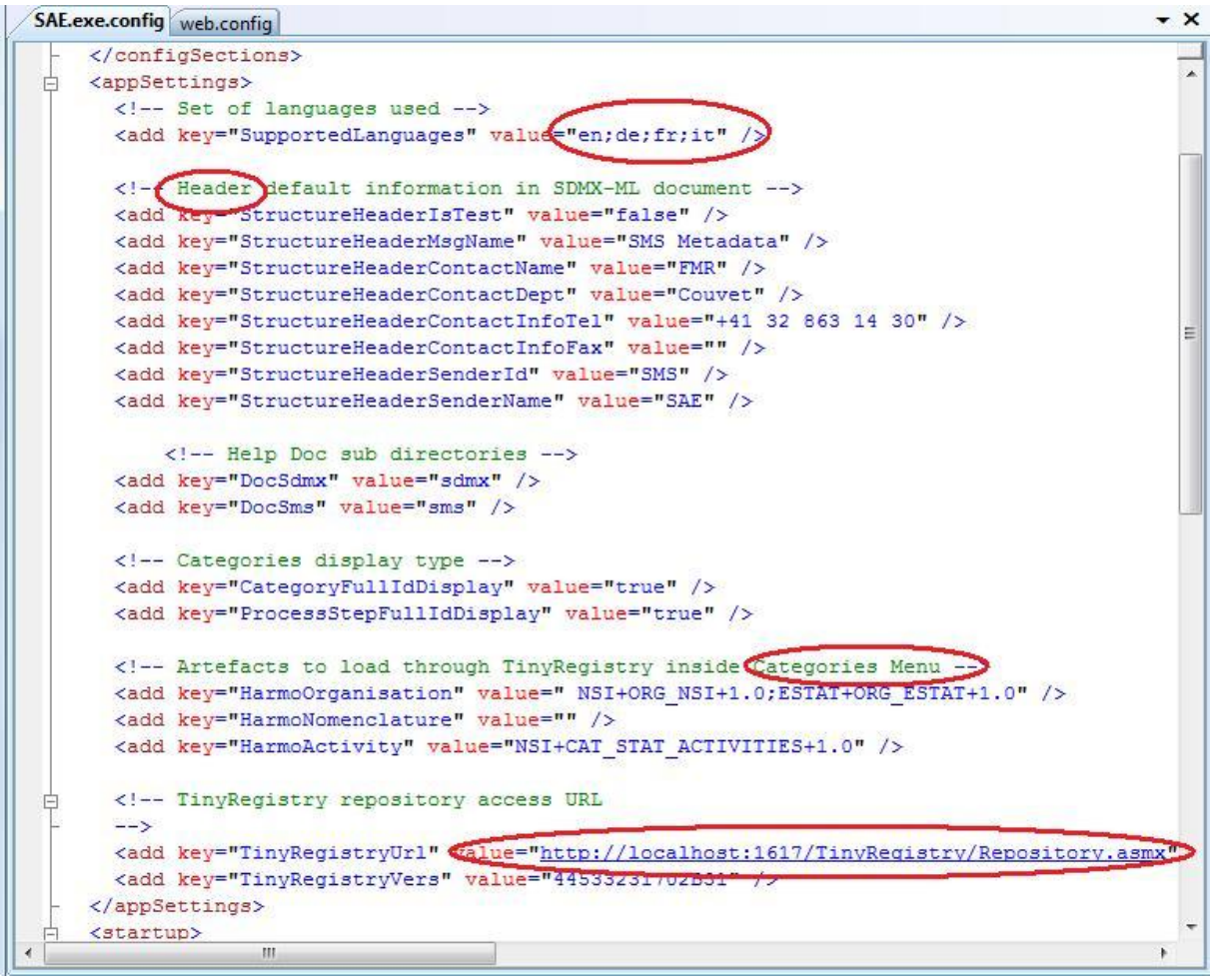


This page displays the list of functionalities available through the TinyRegistry.
Copy the URL of the web site to update the configuration of the SAE editor in order to access the repository from the editor.

Now leave always the repository service running before working with the SAE editor.

4.4 Configure SAE Editor

Edit he file “SAE.exe.config” located in SAE_SRC\SAE_Install\SAE\Editor



1. Ensures that “SupportedLanguages” list matches the one in TinyRegistry config
2. If desired, you can reconfigure some info on generated header for SDMX documents.
3. It is possible to add a direct load from registry for some categories and organisations (available through the Categories menu of SAE)
   The screen sample shows a specific usage.

   Leave the 3 values empty if not used at first: `value=""`

4. Reports the URL copied from the web service page in “TinyRegistryUrl”
   Leaves the value empty if you do not intend to use Repository access.

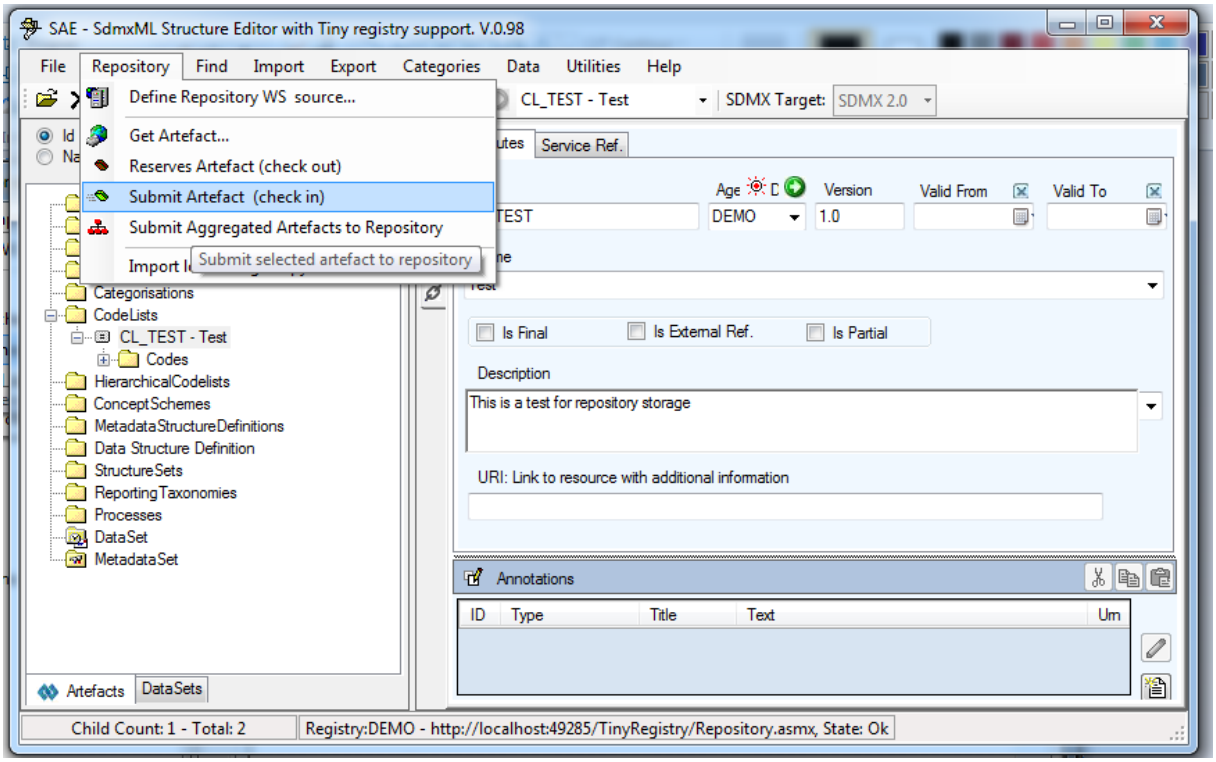
5. Save modified file.

4.5 Test SAE Editor with Repository access

Now Run the Editor and test access to TinyRegistry

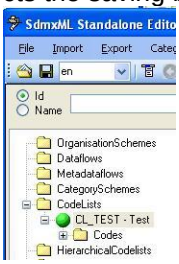
1. Selects CodeLists in Tree

2. Activate popup menu and selects “New Child”
3. Fill Detail screen with ID, AgencyId and Name
4. Open child node and selects Codes
5. Activate popup menu and selects “New Child”
6. Fill Detail screen with CodeValue and OfficialTitle
7. Select the codelist you just created (if you check IsFinal a validity test against schema is done)
8. Choose Menu - Repository – Submit Artefact to Repository

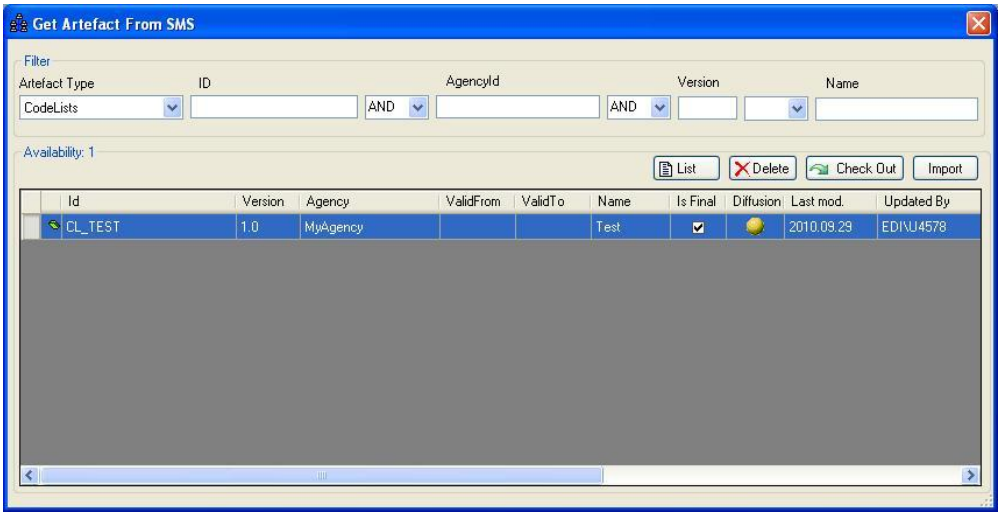


9. Authenticate to Registry.
   Note: Authentication sample is automatic and uses window user

   The tree reflects the saving by displaying a green icon



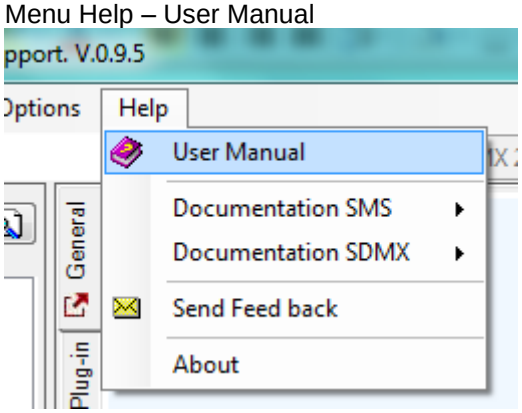
10. Uses Menu Files - Clear All to get an empty tree
11. Activates Menu – Repository – Get Artefact
    A dialog box is displayed.
    Activates the List button
    The list of codelists registered is displayed (last one is the newly)
    (Only one if you selected *Artefacts_empty* folder in web service configuration)



12. Double click on it to retrieve it on editor or
    Activate button “Check Out to reserve it for your usage. In this case you can the place your updated version in Repository with Menu Export – Submit Artefact to Repository.

13. For more details on **SAE Editor, see User manual in Editor SMS directory or in Editor itself**

    Menu Help – User Manual



5 RESTful Web Service description

In April 2011 the SDMX community published a new version of « Guidelines for use of web services ». This version documents the implementation of services based on REST. **(Re**presentational **S**tate **T**ransfer). The documentation is for the version 2.1 of SDMX standard.
This RESTful wrapper is based on this documentation but works currently with SDMX 2.0 artefacts managed in TinyRegistry.

The actual version provides the following ressources:
These are ressources of « Structural Metadata Queries »:

- `datastructure`,
- `metadatastructure`,
- `categoryscheme`,
- `conceptscheme`,
- `codelist`,
- `hierarchicalcodelist`,
- `organisationscheme`,
- `dataflow`,
- `metadataflow`,
- `reportingtaxonomy`,
- `structureset`,
- `process`,

5.1 Parameters used for identifying a resource

:

Parameter	Type	Description
agencyID	A string compliant with the SDMX agencyID	The agency maintaining the artefact to be returned
resourceID	A string compliant with the SDMX common: IDType	The id of the artefact to be returned
version	A string compliant with the SDMX :VersionType	The version of the artefact to be returned

The parameters mentioned above are specified using the following syntax:
http:// ws-entry-point/resource/agencyID/resourceID /version

Furthermore, some keywords may be used:

Keyword	Scope	Description
all[1]	agencyID	Returns artefacts maintained by any maintenance agency[2]
all	resourceID	Returns all resources of the type defined by the resource parameter[2]
all	version	Returns all versions of the resource
latest	version	Returns the latest version in production of the resource[2]

Note :If identifications parameters are missing in the request, the key words all/all/latest are assumed..

[1] As “all” is a reserved keyword in the SDMX RESTful API, it is recommended not to use it as an identifier for agencies, resources or a specific version.
[2] Default, if parameter not specified

5.2 Parameters used to further describe the desired results

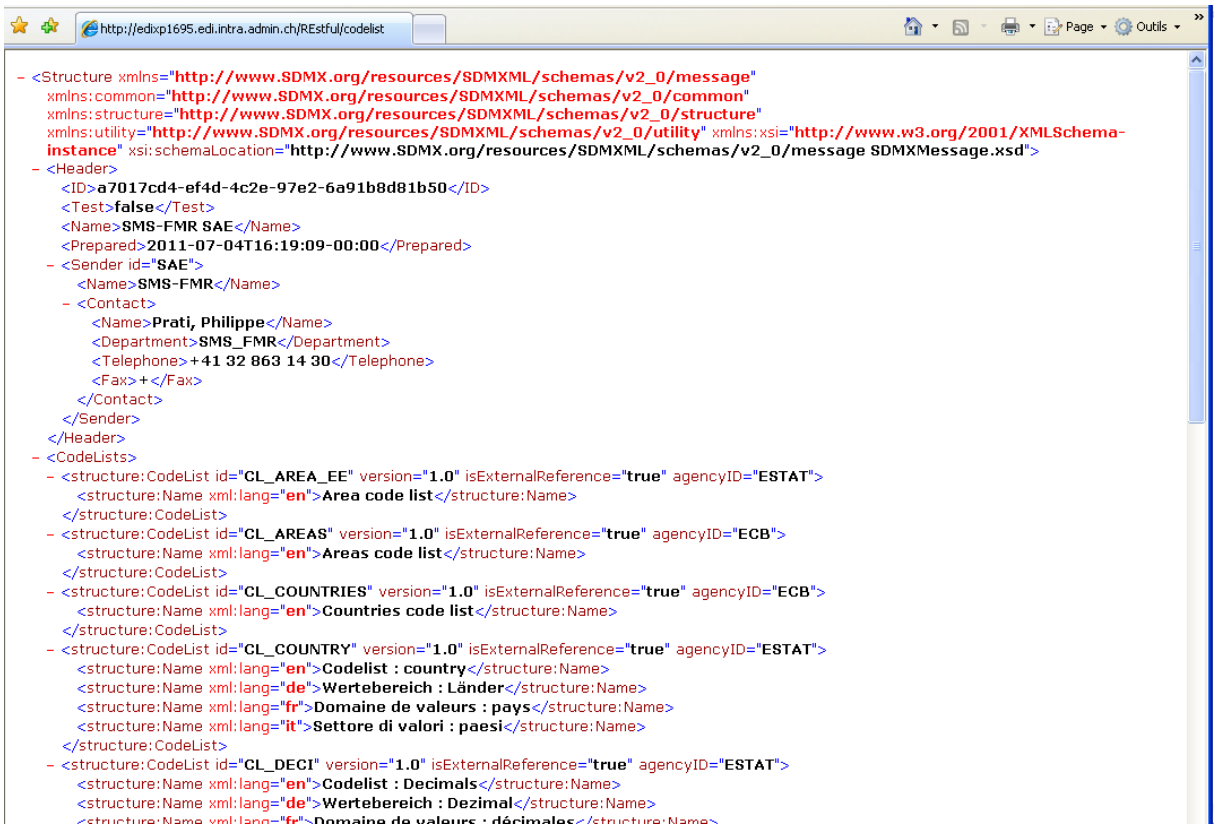
Paramètre de query	Type	Description	Défaut
detail	String	This attribute specifies the desired amount of information to be returned. For example, it is possible to instruct the web service to return only basic information about the maintainable artefact (i.e.: id, agency id, version and name). Possibles values are : • **allstubs** : tout est ramené comme « stub » • **referencestubs** : tous les artéfacts référencés par l'artefact demandé seront en « stub » • **full** : tout le détail est ramené pour tous les artefacts	**full** Except if key word « all » for id **and** agency id. In this last case « **allstubs** » is used as default
references	String	This attribute instructs the web service to return (or not) the artefacts referenced by the artefact to be returned (for example, the code lists and concepts used by the data structure definition matching the query), Supported values aret • **none** : no referenced artefact returned • **children** : direct references, • *descendants : références de références*	**none**

5.3 Request samples

Samples are based to the protocol: *edixp1695.edi.intra.admin.ch*
It has to be changed according to your NSI installation of the Rest service.

- All codelists of all agencies with all versions (in « stub »)
  http://edixp1695.edi.intra.admin.ch/Restful/codelist or
- http://edixp1695.edi.intra.admin.ch/Restful/codelist/all/all/all or
- http://edixp1695.edi.intra.admin.ch/Restful/codelist/all/all/all?detail=allstubs

Answer sample :



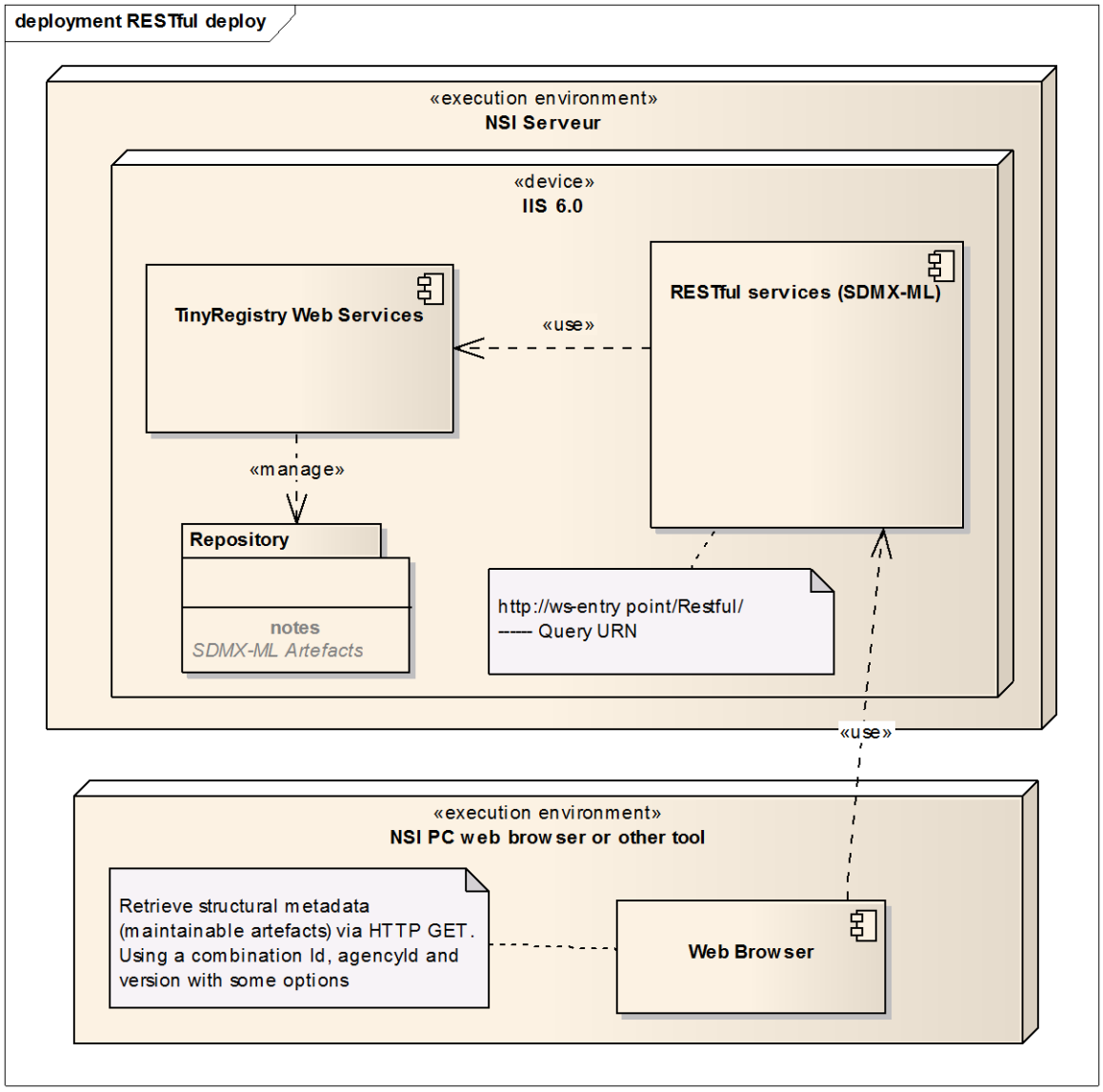

```
- <Structure xmlns="http://www.SDMX.org/resources/SDMXML/schemas/v2_0/message"
    xmlns:common="http://www.SDMX.org/resources/SDMXML/schemas/v2_0/common"
    xmlns:structure="http://www.SDMX.org/resources/SDMXML/schemas/v2_0/structure"
    xmlns:utility="http://www.SDMX.org/resources/SDMXML/schemas/v2_0/utility" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
    instance" xsi:schemaLocation="http://www.SDMX.org/resources/SDMXML/schemas/v2_0/message SDMXMessage.xsd">
  - <Header>
      <ID>a7017cd4-ef4d-4c2e-97e2-6a91b8d81b50</ID>
      <Test>false</Test>
      <Name>SMS-FMR SAE</Name>
      <Prepared>2011-07-04T16:19:09-00:00</Prepared>
    - <Sender id="SAE">
        <Name>SMS-FMR</Name>
      - <Contact>
          <Name>Prati, Philippe</Name>
          <Department>SMS_FMR</Department>
          <Telephone>+41 32 863 14 30</Telephone>
          <Fax>+</Fax>
        </Contact>
      </Sender>
    </Header>
  - <CodeLists>
    - <structure:CodeList id="CL_AREA_EE" version="1.0" isExternalReference="true" agencyID="ESTAT">
        <structure:Name xml:lang="en">Area code list</structure:Name>
      </structure:CodeList>
    - <structure:CodeList id="CL_AREAS" version="1.0" isExternalReference="true" agencyID="ECB">
        <structure:Name xml:lang="en">Areas code list</structure:Name>
      </structure:CodeList>
    - <structure:CodeList id="CL_COUNTRIES" version="1.0" isExternalReference="true" agencyID="ECB">
        <structure:Name xml:lang="en">Countries code list</structure:Name>
      </structure:CodeList>
    - <structure:CodeList id="CL_COUNTRY" version="1.0" isExternalReference="true" agencyID="ESTAT">
        <structure:Name xml:lang="en">Codelist : country</structure:Name>
        <structure:Name xml:lang="de">Wertebereich : Länder</structure:Name>
        <structure:Name xml:lang="fr">Domaine de valeurs : pays</structure:Name>
        <structure:Name xml:lang="it">Settore di valori : paesi</structure:Name>
      </structure:CodeList>
    - <structure:CodeList id="CL_DECI" version="1.0" isExternalReference="true" agencyID="ESTAT">
        <structure:Name xml:lang="en">Codelist : Decimals</structure:Name>
        <structure:Name xml:lang="de">Wertebereich : Dezimal</structure:Name>
```


- All codelists of SDMX agency with all versions (complet)
  http://edixp1695.edi.intra.admin.ch/Restful/codelist/SDMX/all/all
- All codelists of SDMX agency with all versions (en « stubs »)
  http://edixp1695.edi.intra.admin.ch/Restful/codelist/SDMX/all/all?detail=allstubs
- All codelists of ESTAT agency but only last version and as « stubs »
  http://edixp1695.edi.intra.admin.ch/Restful/codelist/ESTAT/all/latest?detail=allstubs
- Detail of Hierarchicalcodelist HCL_SAMPLE of ESTAT in version 2 with détail of referred codelists
  http://edixp1695.edi.intra.admin.ch/Restful/hierarchicalcodelist/ESTAT/HCL_SAMPLE/2.0?references=children
-

-
- All concept schemes of all agencies with all versions (in « stub »)
  http://edixp1695.edi.intra.admin.ch/Restful/conceptscheme

NOTE : If some SDMX document are saved in TinyRegistry as « aggregated documents, The content returned is complete even with option « detail=referencestubs » is used.

5.4 Installation of REST service



5.4.1 Configuration file

The Restful directory is installed in IIS

The web.config file in the restful directory has to be modified to access the NSI deployed TinyRegistry:

The set of languages has to match the one defined in TinyRegistry config.

The resful main directory name used must also be correctly specified.

```
  <appSettings>
    <add key="TinyRegistryUrl"
value="http://localhost:3490/TinyRegistry/Repository.asmx" />
    <add key="TinyRegistryVers" value="44533231702B32" />
    <!-- Set of languages used -->
    <add key="SupportedLanguages" value="en;de;fr;it" />
     <add key="RestDir" value="Restful" />
```

```
<!-- Header default information in SDMX-ML document -->
```

The Rest service has been installed on IIS 5.1 and 6.0 with anonymous connexion.
The configuration procedures are slightly different.

5.4.2 The IIS 5.1 version (NET 2.0)

It requires the following steps:

- Open the website properties in IIS
- Select the Home tab
- Click the Configuration button
- In new window select the Mappings tab
- Click the Add button
- In new window's executable field Browse to the ASP.NET ISAPI DLL (usually something like C:\WINDOWS\Microsoft.NET\Framework\v2.0.50727\aspnet_isapi.dll)
- In the Extension field, type.***** (dot star)
- Unclick the Check that file exists box and click OK

N.B. There is a bug that if your OK button is not active, you need to click the executable field text box, you should see the middle section of the text in the box change from /.../ to the full file path.This should now make the OK button active.

5.4.3 The IIS 6.0 version (NET 2.0)

The fix for IIS 6.0 is to add a "WildCard ISAPI Filter" handler for the site in question. That is,
1- Open IIS, right click, and select Properties for your site.
2- Click on Home Directory
3- Click on the Configuration button
4- Add a WildCard application map to the bottom window:
    a- click Insert
    b- Browse to windows\microsoft.net\framework\your-version\
    c- Add aspnet_isapi.dll
    d- Make sure "Verify that file exists" is UNchecked
5- Save everything
6- RESTART IIS <- important