



## NextGRID Data Storage Profile

<b>Editors:</b>	Clive Davenhall	UEDIN

Date	Author	Comments	Version	Status
6 Aug. 2007	Clive Davenhall	Incomplete draft	0.1	Draft
7 Aug. 2007	Clive Davenhall	Preliminary draft for comments	0.2	Draft
10 Sept. 2007	Clive Davenhall	Draft for comments	0.3	Draft
9 Oct. 2007	Clive Davenhall	Modified follow review	0.4	Draft
10 Dec. 2007	Clive Davenhall	Clarified the relation to the registry profile.	0.5	Draft
25 Feb. 2008	Clive Davenhall	Clarified the relation between data storage and data access.	0.6	Draft
1 Mar. 2008	Clive Davenhall	Final edits.	0.7	Draft
11 Mar 2008	D. Snelling	Final QA	1.0	Final



**1 INTRODUCTION..... 3**

1.1 Profile Overview ..... 3

1.2 Relationships to Other Profiles..... 3

1.3 Relationship to Use Cases..... 4

    1.3.1 Terminology..... 4

1.4 Notational Conventions ..... 4

1.5 Profile Identification and Versioning..... 5

**2 PROFILE CONFORMANCE ..... 5**

2.1 Conformance Targets ..... 5

2.2 Claiming Conformance ..... 6

**3 DATA STORAGE FUNCTIONS..... 6**

3.1 NextGRID Datasets ..... 6

3.2 Data Storage Functionality ..... 6

3.3 Data Storage Implementation..... 7

**4 NEXTGRID BASIC PROFILE ..... 7**

**5 NEXTGRID ACCESS PROFILE ..... 7**

**6 NEXTGRID REGISTRY PROFILE ..... 8**

6.1 Contents of a Store’s Inventory ..... 8

6.2 Identifying a File in a Storage Service ..... 8

6.3 Annotating a File in a Storage Service..... 9

6.4 Specifying a File to the Storage Functions..... 11

6.5 Data Transport..... 12

**7 ACCESS AND LIFETIME MANAGEMENT ..... 12**

**8 REFERENCES..... 13**



## 1 Introduction

This document defines the NextGRID Data Storage Profile 1.0 (hereafter, “this Profile”), consisting of a set of de facto, institutional or evolving institutional Web Services specifications, along with clarifications, refinements, interpretations and amplifications of those specifications that promote interoperability among implementations of those specifications.

Section 1 introduces this Profile, and explains its relationships to other profiles.

Section 2, “Profile Conformance”, explains what it means to be conformant to this Profile.

Subsequent sections present the requirements for a NextGRID data storage service and specify the Storage Profile.

This Profile forms part of the NextGRID Generalized Specifications, which aim at capturing NextGRID architectural concepts in a set of profiles that may be composed together. These profiles are specified in such a way that they could be implemented in terms of other well known specifications. While overall consistency is achieved at the conceptual level, and captured through the motivating Use Cases accompanying each Generalized Specification, system implementations, which may be based on other specifications, may not be consistent with this Profile. Thus, each profile defines a realisation of the underlying concept that can be implemented. However, implementers of the NextGRID architecture may need to support multiple underlying specifications.

### 1.1 Profile Overview

This Profile is intended for use when implementing data storage services that are in line with the concepts of NextGRID [1]. It mandates adherence to a certain set of specifications and clarifies their use. A service implementation that uses those specifications in a manner conformant with this Profile may be said to be an “implementation of the NextGRID Data Storage Profile 1.0” or, informally, to be a “NextGRID Data Storage Service”, a “data store” or simply a “store.”

The primary issues addressed in this Profile are as follows:

- The specification of the basic functionality of a NextGRID Data Storage Service.
- The relation between the NextGRID Data Storage and Data Access Profiles.

### 1.2 Relationships to Other Profiles

This Profile extends the *NextGRID Basic Profile* 1.0 [2].

#### NextGRID Data Access Profile

This Profile complements the *NextGRID Data Access Profile* [3].

## NextGRID Registry Profile

For a data storage service which stores files, access to the list of files stored should conform to the *NextGRID Registry Profile* [4].

## NextGRID Naming Profile

Service naming conforms to the *NextGRID Naming Profile* [5].

## NextGRID Data Transfer Profile

To input or output data from a data storage service, the service needs to be able to act as either a sink or a source of data within the context of the *NextGRID Data Transfer Profile* [6].

## NextGRID SLA Schema

Service Level Agreements (SLAs) between a data storage service and other NextGRID components use the *NextGRID SLA Schema* [7].

### 1.3 Relationship to Use Cases

This Profile represents a mapping from the use cases and requirements described in the *NextGRID Data Storage and Access Use Cases and Requirements* document [8] to the Web Services Resource Framework (WSRF) technology [9]. A data storage service implements the *NextGRID Basic Profile* [2]. It provides access to data in conformance with the *NextGRID Data Access Profile* [3] and aspects of its functionality may conform to the *NextGRID Registry Profile* [4].

#### 1.3.1 Terminology

This Profile adopts the terminology of the *NextGRID Data Storage and Access Use Cases and Requirements* [8]. See that document, particularly Section 1.3 ‘Terminology’, for descriptions of the terms pertaining to data processing used in this Profile. Hereafter terms defined in that section of that document are shown in **bold** in this Profile. See also the OGSA *Glossary of Terms* [10].

### 1.4 Notational Conventions

The keywords "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC2119 [11].

Normative statements of requirements in this Profile are presented in the manner detailed in the *WS-I Basic Profile* 1.1 [12], Section 2.1, “Conformance Requirements”.

Both requirement statements and extensibility statements can be considered namespace-qualified. This specification uses a number of namespace prefixes throughout; their associated URIs are listed below. Note that the choice of any namespace prefix is arbitrary and not semantically significant.

**Table 1 Namespaces used by NextGRID Data Storage Profile 1.0.**

Prefix	Namespace
xsd	http://www.w3.org/2001/XMLSchema
wSDL	http://schemas.xmlsoap.org/wSDL
wsa	http://www.w3.org/2005/08/addressing
ng-store	http://www.nextgrid.org/datastorageprofile/v-1

## 1.5 Profile Identification and Versioning

Profile identification and versioning uses the style described in the *WS-I Basic Profile 1.1* [12] and abides by the normative descriptions contained therein. The name of this Profile is “NextGRID Data Storage Profile” and the version number is “1.0”.

## 2 Profile Conformance

Conformance to this Profile is defined normatively in the *WS-I Basic Profile 1.1* [12], Section 2, “Profile Conformance”. This Profile abides by those definitions.

### 2.1 Conformance Targets

Since this Profile is an extension of the *NextGRID Basic Profile 1.0* [2], which itself extends the *WS-I Basic Profile 1.1* [12] and the *OGSA WSRF Basic Profile 1.0* [13], it may place further restrictions on conformance targets defined therein.

The following conformance targets are used in this Profile:

- **DESCRIPTION** – descriptions of types, messages, interfaces and their concrete protocol and data format bindings, and the network access points associated with Web services (e.g., WSDL descriptions) (from *WS-I Basic Profile 1.1*).
- **STORE** – An instance of an implementation of a NextGRID data storage service; such an instance necessarily implements a wsdl:port (from *WS-I Basic Profile 1.1*, without “bindingTemplate” from the namespace urn:uddi-org:api\_v2).

- **FILESTORE** – A STORE which contains **files**.
- **INVENTORY** – For a STORE which operates on **files**: a list of all the **files** the STORE contains. For each **file** its name and other metadata are listed.

## 2.2 Claiming Conformance

Claims of conformance to this Profile and the attachments mechanisms are the same as normatively described in the WS-I *Basic Profile* 1.1 [12].

The conformance claim URI for this Profile is “<http://www.nextgrid.org/datastorageprofile/v-1>”.

## 3 Data Storage Functions

### 3.1 NextGRID Datasets

A **dataset** in the context of NextGRID is ‘any identifiable collection of data; that is, one which can be uniquely and independently identified by a NextGRID name (see the *NextGRID Naming Profile* [5]) or referenced with an EPR.’ The *NextGRID Data Storage and Access Use Cases and Requirements* [9], Section 1.3 and 1.4, pp4-5 gives further details.

**Datasets** may be instantiated in more than one way. Specifically, they may be **files** or **selections**:

- A **file** is a **dataset** instantiated as a file in some operating system.
- A **selection** is a **dataset** instantiated as a set of tuples in some database which satisfy some criterion. A **selection** may be: (i) a whole database, (ii) a table in a database, or (iii) a selection of tuples from one or more tables in a database (that is, a view).

### 3.2 Data Storage Functionality

**Data storage** is the persistence of data through time. A NextGRID **data storage service**, or a **store**, superintends the set of **datasets** which it is storing. A **store** will provide the following functionality:

- Enforce lifetime management. In the absence of manipulation by a suitably authorised **client** it will maintain **datasets** for their specified lifetime and then dispose of them in accordance with the SLA [7] between the **store** and the **dataset**’s **publisher**.
- Provide **access** to **datasets** in conformance with the *NextGRID Data Access Profile* [3].
- Enforce any access restrictions that may have been prescribed in the SLA [7] between the **store** and the **dataset**’s **publisher**.

Recall that **datasets** may be **files** or **selections** in a database. If the **store** is hosting a database from which **selections** may be extracted then the **store** will provide sufficient metadata to allow the

database to be queried (typically the names of tables and their attributes, query languages supported *etc*).

If the **store** is hosting a collection of **files** then it must maintain an inventory of the **files** that it contains. This inventory must conform to the *NextGRID Registry Profile* [4], with additional rules to allow the names of the files to be identified.

It is also possible for an external NextGRID registry, which is not part of the **store**, to maintain its own list of some or all of the **files** in a **store**. Such an external registry can provide indirect access to the **files** in a **store**, but does not otherwise affect the operation of the **store**.

### 3.3 Data Storage Implementation

This Profile prescribes aspects of a WSRF interface to a **data storage service**. It does not prescribe how such a service can be implemented. Often a **data storage service** will be implemented as a Web service interface to a DBMS or an underlying file system, as appropriate. However, there are other possibilities. All the functionality of the **data storage service** might be built into the Web service code or the **store** might provide a portal to some other, perhaps distributed system. The details of the implementation are out of scope of this Profile.

## 4 NextGRID Basic Profile

This section of the Profile incorporates the following specification by reference:

- *NextGRID Basic Profile* 1.0 [2].

All requirements stated in the *NextGRID Basic Profile* 1.0 must be observed by implementations of this Profile.

## 5 NextGRID Access Profile

This section of the Profile incorporates the following specification by reference:

- *NextGRID Data Access Profile* [3].

One of the functions of a **store** is to provide **access** to the **datasets** that it contains. Such **access** should conform to the *NextGRID Data Access Profile*. Consequently:

**R0501** A STORE MUST provide access to a dataset that it contains in conformance with the NextGRID Data Access Profile [3].

## 6 NextGRID Registry Profile

This section of the Profile incorporates the following specification by reference:

- *NextGRID Registry Profile 1.0* [4].

This Profile incorporates the *NextGRID Registry Profile 1.0*. **Stores** that contain **files** must maintain an inventory of all the **files** that they contain. Such an inventory must conform to the *NextGRID Registry Profile*, except for the extensions and amendments noted below.

### 6.1 Contents of a Store's Inventory

A NextGRID registry [4] can contain an arbitrary list of items. However, the inventory of a **store** which contains **files** comprises by definition the list of the **files** in the **store**. The **store** must maintain the correspondence between the **files** that it is storing and the entries in its inventory. Consequently:

**R0611** A FILESTORE MUST maintain an INVENTORY comprising the **files** that it contains.

**R0612** The INVENTORY for a FILESTORE MUST contain one (and only one) entry for each **file** currently included in the FILESTORE.

Or equivalently:

**R0613** A FILESTORE MUST maintain a one-to-one correspondence between the entries in its INVENTORY and the **files** that it contains. This correspondence MUST be maintained throughout all the operations performed by the FILESTORE.

### 6.2 Identifying a File in a Storage Service

A **data storage service** which stores **files** maintains a set of **files** and allows them to be retrieved. **Files** in **data stores** are identified by a unique identifier, either a NextGRID name [5] or an EPR. The NextGRID name is a specialisation of the WS-Name, which itself extends WS-Addressing (see [5] for details and references).

Every **file** in a **store** has an entry in the **store's** inventory. The metadata for this entry must contain the name of the **file** in an identifiable fashion. The <ng-store:DatasetName> element is defined for this purpose. Consequently:

**R0621** The metadata for each **file** included in an INVENTORY MUST contain one (and only one) <ng-store:DatasetName> element.

The `<ng-store:DatasetName>` element has a `<wsa:EndpointReference>` as a child element. The `<wsa:EndpointReference>` element can, of course, contain WS-Naming elements. The use of the `<wsa:EndpointReference>` and WS-Naming elements are restricted as described in the *NextGRID Naming Profile* [5].

Thus, the structure of the `<ng-store:DatasetName>` element is as follows.

```
<ng-store:DatasetName>
  <wsa:EndpointReference >
    ...
  </wsa:EndpointReference >
</ ng-store:DatasetName >
```

Further:

**R0622** *The `<ng-store:DatasetName>` element present in each entry in the INVENTORY MAY optionally contain additional child elements, to permit future extensibility.*

No such child elements of `<ng-store:DatasetName>` are considered in this version of this Profile.

### 6.3 Annotating a File in a Storage Service

This Profile does not constrain the metadata which may annotate the entry for each **file** in a **store's** inventory, other than as noted in Section 6.2, above. However, it suggests the following elements for storing commonly occurring metadata. The first four of these elements deliberately have the same names as the equivalent properties in the *OGSA ByteIO* specification [14]. Though these elements are optional, if they are used, they must be used as described in this Profile.

Table 2 Optional elements to annotate entries in an inventory.

Name	xsd type	Description	In ByteIO?
Size	nonNegativeInteger	Size of the file in bytes.	✓
CreateTime	dateTime	The timestamp for when this file was created.	✓
ModificationTime	dateTime	The timestamp for the most recent occasion on which this file was modified.	✓
AccessTime	dateTime	The timestamp for the most recent occasion on which this file was accessed.	✓
ExpiryTime	dateTime	The timestamp at which the lifetime of this file is scheduled to end.	

Thus:

**R0631** The metadata for each **file** included in an INVENTORY MAY contain one (and only one) `<ng-store:Size>` element.

**R0632** The metadata for each **file** included in an INVENTORY MAY contain one (and only one) `<ng-store:CreateTime>` element.

**R0633** The metadata for each **file** included in an INVENTORY MAY contain one (and only one) `<ng-store:ModificationTime>` element.

**R0634** The metadata for each **file** included in an INVENTORY MAY contain one (and only one) `<ng-store:AccessTime>` element.

**R0635** The metadata for each **file** included in an INVENTORY MAY contain one (and only one) `<ng-store:ExpiryTime>` element.

**R0636** If any of the `<ng-store:Size>`, `<ng-store:CreateTime>`, `<ng-store:ModificationTime>`, `<ng-store:AccessTime>` or `<ng-store:ExpiryTime>` elements are used in the metadata for a file included in an INVENTORY, they MUST be used as described in this Profile.

The following fragment of XML illustrates the use of these elements.

```
<ng-store:Size>1230000<ng-store:Size>  
<ng-store:CreateTime>2008-02-01T10:00:00<ng-store:CreateTime >  
<ng-store:ModificationTime>2008-03-12T10:00:00<ng-store:ModificationTime >  
<ng-store:AccessTime>2008-03-12T10:03:14<ng-store:AccessTime >  
<ng-store:ExpiryTime>2008-04-01T00:00:01<ng-store:ExpiryTime >
```

## 6.4 Specifying a File to the Storage Functions

**Files** in a **file store** can be operated on by the functions: **publish**, **update**, **delete** and **retrieve**. The following rules prescribe the unambiguous identification of the **files** on which these functions operate. To permit implementation efficiencies each function may operate on a single **file** or a list of **files**.

- R0641** *The section of the FILESTORE DESCRIPTION pertaining to the invocation of the **publish** function MUST contain one or more <ng-store:DatasetName> elements. Each element MAY suggest an EPR to reference the corresponding **file** when it is ingested into the FILESTORE.*
- R0642** *The section of the FILESTORE DESCRIPTION pertaining to the response of the **publish** function MUST contain one or more <ng-store:DatasetName> elements. Each element MUST contain a valid EPR by which the corresponding **file** can be referenced. There MUST be a one-to-one correspondence between the <ng-store:DatasetName> elements in the response with those in the corresponding invocation.*
- R0643** *The section of the FILESTORE DESCRIPTION pertaining to the invocation of the **update** function MUST contain one or more <ng-store:DatasetName> elements. Each element MUST contain a valid EPR by which the corresponding **file** can be referenced.*
- R0644** *The section of the FILESTORE DESCRIPTION pertaining to the invocation of the **delete** function MUST contain one or more <ng-store:DatasetName> elements. Each element MUST contain a valid EPR by which the corresponding **file** can be referenced.*
- R0645** *The section of the FILESTORE DESCRIPTION pertaining to the invocation of the **retrieve** function MUST contain one or more <ng-store:DatasetName> elements. Each element MUST contain a valid EPR by which the corresponding **file** can be referenced.*

## 6.5 Data Transport

A **data storage service** provides **access** to the **datasets** that it is storing in conformance with the *NextGRID Data Access Profile* [3]. Consequently, the constraints on data transport [6] during **access** read, write and update operations (see the *NextGRID Data Access Profile*, Section 7, p14) also apply to this Profile. Specifically:

**R0651** When a **dataset** is being read from a **STORE** the **STORE MUST** act as a source of data within the context of the NextGRID Data Transfer Profile [6].

**R0652** When a **dataset** is being written to, or updated on, a **STORE** the **STORE MUST** act as a sink of data within the context of the NextGRID Data Transfer Profile [6].

## 7 Access and Lifetime Management

A **store** is responsible for guaranteeing the integrity of the **datasets** that it contains, in conformance with the SLA [7] that it has agreed with the **dataset's publisher** and subject to any subsequent modification by a suitable authorised **client**. Specifically:

**R0701** A **STORE MUST** enforce any access restrictions on a **dataset** that were included in the SLA [7] between the **STORE** and the **dataset's publisher**.

**R0702** In the absence of manipulation by a suitably authorized **client**, a **STORE MUST** retain any **dataset** that it contains until the **dataset's** expiry date is reached.

**R0703** When the expiry date of a **dataset** is reached the **STORE** holding the **dataset** **MUST** dispose of the **dataset** in accordance with the SLA [7] between the **STORE** and the **dataset's publisher**.

## 8 References

- [1] D. Snelling, M. Fisher, A. Basermann, F. Wray, P. Wieder and M. Surridge, *NextGRID White Paper*, v0.5, NextGRID Project, 25 July 2005, [http://www.nextgrid.org/download/publications/NextGRID\\_Architecture\\_White\\_Paper.pdf](http://www.nextgrid.org/download/publications/NextGRID_Architecture_White_Paper.pdf)
- [2] V. Li, D. Snelling and A. Anjomshoaa (eds.), *NextGRID Basic Profile*, V1.0, NextGRID Project, 2 January 2008.
- [3] C. Davenhall, *NextGRID Data Access Profile*, v0.7, NextGRID Project, 1 March 2008.
- [4] P. Hasselmeyer, M. Surridge and P. Wieder, *NextGRID Registry Profile*, V1.0, NextGRID Project, 5 February 2008.
- [5] S. Davey and D. Snelling, *NextGRID Naming Profile*, V1.0, NextGRID Project, 26 February 2008.
- [6] M. Drescher, S. Davey and A. Anjomshoaa, *NextGRID Data Transfer Profile*, v1.0, NextGRID Project, 19 February 2008.
- [7] D. Snelling, *NextGRID SLA Schema*, v0.2, NextGRID Project, 7 July 2007.
- [8] C. Davenhall, *NextGRID Data Storage and Access Use Cases and Requirements*, v1.0, NextGRID Project, 11 March 2008.
- [9] T. Banks. *Web Services Resource Framework (WSRF) – Primer v1.2*, Committee Draft 02, Organization for the Advancement of Structured Information Standards, 26 May 2006. <http://docs.oasis-open.org/wsrp/wsrp-primer-1.2-primer-cd-02.pdf>
- [10] J. Treadwell, GFD-I.120: *Open Grid Services Architecture Glossary of Terms*, V1.6, Open Grid Forum, 12 December 2007.
- [11] S. Bradner, RFC 2119: *Key words for use in RFCs to Indicate Requirement Levels* (the Internet Engineering Task Force Best Current Practice), March 1997. <http://www.ietf.org/rfc/rfc2119>
- [12] K. Ballinger, D. Ehnebuske, C. Ferris, M. Gudgin, C.K. Liu, M. Nottingham, and P. Yendluri (eds.): *Basic Profile* Version 1.1, Web Services Interoperability Organization Final Material, 24 August 2004. <http://www.ws-i.org/Profiles/BasicProfile-1.1.html>
- [13] I. Foster, T. Maguire, D. Snelling, *OGSA™ WSRF Basic Profile* 1.0, GFD-R-P.072, Open Grid Forum, May 1, 2006, <http://www.ogf.org/documents/GFD.72.pdf>
- [14] M. Morgan, *ByteIO Specification*, V1.0, GFD-R-P.087, Open Grid Forum, 31 October 2006, <http://www.ogf.org/documents/GFD.87.pdf>