SPUD SERVICES USER GUIDE

This document describes the use of SPUD web services for submitting, retrieving and managing data contained within the IRIS SPUD System

Revision History

Version	Date	Revision Description
1.0	8/17/2011	Initial Revision
1.1	8/31/2011	Additions
1.2	9/22/2011	Media Services

1. Introduction

This document describes the usage of SPUD's web service interface (aka SPUD Services), its protocols and data structures. These services are not to be confused with the SPUD website (www.iris.edu/spud) which is a separate beast altogether, though it makes use of a subset of SPUD Services for media access from within SPUD web pages.

The services described here are used to manage instances of SPUD products (items), and their attendant data which are generally visible via the SPUD website. SPUD Services allow the management (i.e. creation, reading, updating and deleting) of items and data via standard HTTP protocols using a RESTful interface.

2. Overview

Terminology

Any discussion of SPUD data will use various terms, e.g. *item*, *contribution,* whose meanings are quite specific (or not) within the SPUD context. A brief outline of some of the more notable terms is provided below.

Essentially, SPUD displays and SPUD services manage **Items** of various **Product Types** ('GMV', 'Event Plot', etc.) that have metadata and data. **Item meta-data** are data that reflect aspects of the item that can be queried against, e.g. event location, provenance. Items can also contain actual data (Movies, Images, SEED files), which are defined as one of **Primary Data, File Data, Attachments** or **Source Data** depending on the data's type and classification. Furthermore, Primary Data objects can also have **Alternate Data,** classified as either **Screens** or **Thumbnails** (both typically smaller static representations of a larger data object), which can be substituted for the original data when a larger representation is not required.

These terms are further described in the table below.

Term	Definition
Item	The basic SPUD data unit. Viewable in SPUD.
Contribution	An individual submission of an item to SPUD
Active / Inactive	Visibility state of a submission in the SPUD website. Active items are presented in SPUD queries.
Item (Meta) Data	The meta-data of an item stored in SPUD's database, usually represented as XML.
Product Type	An item type, e.g. 'GMV', 'Event Plot', Calibration'
Sub Type	The item product sub type, e.g. "Type2_Calibration"

Primary Data	Various data (media, whatnot) associated with an item, stored in the SPUD database.
Source Data	The source data (typically XML) used in the submission of the item (stored in the SPUD database)
Alternate Data	Various data associated with a primary data object of an item (stored in the SPUD database). Generally used for display purposes in SPUD.
File Data	Various data, usually or considerable size, associated with an item stored on the server's file system.

Uses of SPUD Services

SPUD data product managers submit their items into the SPUD system using SPUD HTTP Services. Via these services it is also possible to overwrite, delete and augment items existing within the SPUD system.

SPUD Services are used to access item metadata, item data (images, movies, etc.) and other data related services e.g. moment tensor focal mechanism plots.

SPUD Services can also be used to access information regarding the holdings of the SPUD system, e.g. the number of 'Active' GMV items, SPUD IDs, etc.

Finally, SPUD Services can be used to access specific utilities within the SPUD system, e.g. accessing ZIP file bundles of items.

HTTP Methods

SPUD Services use four standard HTTP methods in the management of SPUD data. URL structure will be discussed below.

HTTP Method	Operation
POST	Creates a new object
GET	Reads a single object or list of objects
PUT	Updates (Replaces) an existing object
DELETE	Deletes an existing object

Examples

To access the item meta-data for a single GMV item with ID 1234:

GET www.iris.edu/spudservice/gmv/1234

To access a list of all moment tensor holdings (Note lack of specified ID in URL):

GET www.iris.edu/spudservice/momenttensor

To create a new Earth Model item (no ID specified in URL)

POST www.iris.edu/spudservice/earthmodel

To replace an existing Earth Model with SPUD ID 2345:

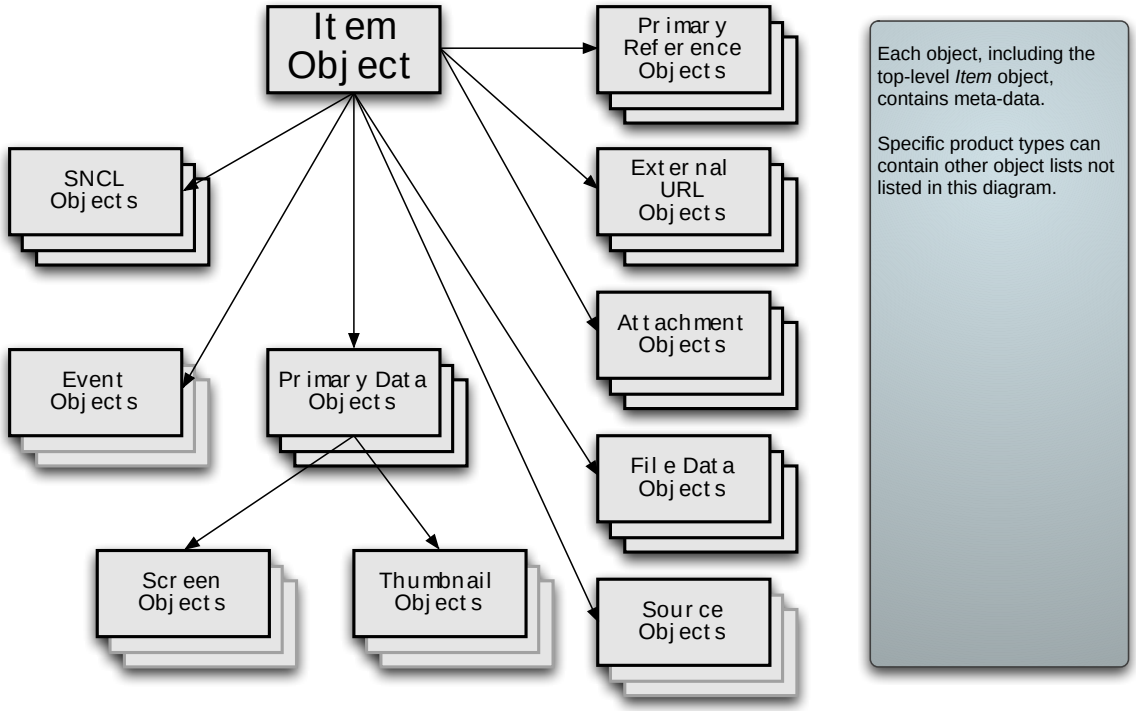
PUT www.iris.edu/spudservice/earthmodel/2345

To delete an existing Earth Model with SPUD ID 2345:

DELETE www.iris.edu/spudservice/earthmodel/2345

3. Item Structure

SPUD Items are organized in an object hierarchy as described in the figure below. An Item will be of a specific *product type*, e.g. GMV, Earth Model, etc., however the diagram below describes a generic SPUD item and includes only the common objects associated with all SPUD items. Item of different product types will generally include product specific meta-data as well as other object lists as needed in the implementation of the particular product type.



Object Diagram of Common SPUD Item Structure

SPUD Object Descriptions

The table below provides brief descriptions and standard usage for component objects of a SPUD item.

Object	Description
Item	Basic Item object. Contains sub-objects as well as product-generic meta-data about this item / submission, including product location / time, provenance, product type, sub type, etc.
Primary Data	Storage object for standard SPUD data stored within the SPUD database. May contain sub-objects of types Screen or Thumbnail which are reduced size objects suitable for web-based display
Screen	Storage object for screen resolution version of a Primary Data object.†
Thumbnail	Storage object for thumbnail resolution version of a Primary Data object.†
File Data	Storage object to file-storage based data in SPUD. Generally used when the data is of unwieldy size.
Attachment	Storage object for a generic data object. Typically used to store attendant data for an item which is not displayed directly via the SPUD website.
Source Data	The original source data (generally XML) used in the creation of this item submission.
Primary Reference	A URL of specific significance to this item and receiving special display in the SPUD web site.†
External URL	A URL associated with this item.
Event	Event object associated with the item. Not required for all product types.†
SNCL	SNCL object associated with the item. Not required for all product types.

† Although it is possible to associate multiple objects of this type to its parent object, typically only one is used within the SPUD system.

Representational URLs and URL Structure

The URLS used in SPUD services are representational in that the form of these URLs represents the hierarchy and structure of the objects that are to be acted upon.

In general, a SPUD service URL is constructed in the following manners:

Item List	SERVICE_PATH/SPUD_SERVICE/ITEM_TYPE
Item	SERVICE_PATH/SPUD_SERVICE/ITEM_TYPE/ITEM_ID
Object List	SERVICE_PATH/SPUD_SERVICE/ITEM_TYPE/ITEM_ID/OBJECT_TYPE
Object	SERVICE_PATH/SPUD_SERVICE/ITEM_TYPE/ITEM_ID/OBJECT_TYPE/OBJECT_ID
Sub Object List	SERVICE_PATH/SPUD_SERVICE/ITEM_TYPE/ITEM_ID/OBJECT_TYPE
Sub Object	SERVICE_PATH/SPUD_SERVICE/ITEM_TYPE/ITEM_ID/OBJECT_TYPE/OBJECT_ID/SUB_TYPE/SUB_ID

Where:

SERVICE_PATH is typically `www.iris.edu`

SPUD_SERVICE is `spudservice` (or `spudservicebeta`)

ITEM_TYPE is from the set [`gmv`, `eventplot`, `filmchip`, etc. plus `item`]

ITEM_ID is the numeric SPUD ID of a particular item

OBJECT_TYPE is from the set [`source`, `primarydata`, `filedata`, `attachment`, `event`, `externalurl`, `primaryreference`]

OBJECT_ID is the number SPUD ID of a particular object attached to an item.

SUB_TYPE is from the set [`screen`, `thumbnail`]

SUB_ID is the numeric SPUD ID of a particular sub-object attached to an object.

Generally, if a URL does not end in a number, i.e. an ID, then it operates on a list of objects. If it does end with a number it operates on a single object.

Use of the Generic Product Type *Item*

The *generic* product type string "item" can be used in place of a specific product type string. This causes SPUD services to be unrestricted as to product type. So whereas a GET of www.iris.edu/spudservice/gmv/55 will only return a GMV item with ID 55 (if it exists), a GET of www.iris.edu/spudservice/item/55 will return an item of any product type with ID 55 (again, if it exists).

This is particularly useful in queries that return lists of items based on specific query parameters.

Examples

If one wanted to create a new Primary Data object for a GMV with ID 456 one would POST Primary Data (XML and object data) to the URL:

www.iris.edu/spudservice/gmv/456/primarydata

And if one wished to delete thumbnail 88 associated with Primary Data object 67 of GMV 23 one would use this URL:

www.iris.edu/spudservice/gmv/23/primarydata/67/thumbnail/88

Valid Item Service URLs

The tables below list valid URLs for accessing items and sub-object meta-data broken down by HTTP method. This is not an exhaustive list for all SPUD services, it only illustrates item meta-data services, and only for generic cross-product services. Specific product types may support additional services not listed below.

The URL does not include either the server path or the spudservice path component.

Product Types in URLs

Note that the table below uses the product type GMV, but any valid SPUD product type can be used

URL	GET Accesses
gmv	Item meta-data for all GMVs in list format
gmv/id	Item meta-data for GMV with ID=id
gmv/id/primarydata	Primary Data meta-data for all Primary Data associated with GMV with ID=id
gmv/id/primarydata/pid	Primary Data meta-data for Primary Data object with ID=pid associated with GMV with ID=id
gmv/id/primarydata/pid/thumbnail	Thumbnail meta-data for all Thumbnail objects associated with Primary Data object with ID=pid associated with GMV with =ID
gmv/id/primarydata/pid/screen	Screen meta-data for all Screen objects associated with Primary Data object with ID=pid associated with GMV with ID=id
gmv/id/primarydata/pid/thumbnail/tid	Thumbnail meta-data for Thumbnail

	object with ID=tid associated with Primary Data object with ID=pid associated with GMV with ID=id
gmv/id/primarydata/pid/screen/sid	Screen meta-data for Screen object with ID=tid associated with Primary Data object with ID=pid associated with GMV with ID=id
gmv/ID/source	Source data used to generate GMV item with ID=id
gmv/ID/event	Event meta-data for all Event objects associated with GMV with ID=id[1]
gmv/ID/event/eid	Event meta-data for Event object with ID=eid associated with GMV with ID=id
gmv/ID/provenance	Provenance meta-data for GMV with ID=id
gmv/ID/locationtime	Product location and time information for GMV with ID=id

Valid Data Service URLs

The table below lists valid URLs for accessing streaming data from SPUD data objects. These services differ from Item Services in that they return a stream of data with the appropriate content type (MIME type) rather than an XML representation of the accessed object.

The URLs below do not include either the server path or the spudservice path component.

URL	GET Accesses
data/id	Streamed content of Primary, Attachment or Source data object with ID=id
alternate/id	Streamed content of Thumbnail or Screen object with ID=id

[1] SPUD items generally have a single Event object.

filedata/id	Streamed content of File Data object with ID=id

Other Services

Various other services exist which fulfill various needs. They are described below.

4. Item Services

Item services are used to retrieve, submit and manage Item data. In general the data formats accepted are XML or, in the case of item submission or replacement, an XML document and other attendant data files. Data services, which return data objects associated with an item, e.g. images, movies, are described below.

Metadata Representation

Item services accept and return item metadata in XML format. Services returning individual items return an XML representation of the full item including all metadata and attached data objects, e.g. Primary Data etc.

Services that return multiple items return a *thin* XML representation of the item's metadata, which is generally a subset of the items' entire metadata set. This is done to reduce the amount of returned data.

Metadata Structure

The SPUD XML structure is extremely straightforward and has few constraints. It is specified within the SPUD system and the meta-data schema is currently not published. (XSD available upon request. Please allow 4-6 weeks for delivery. Handling charges may apply). However, examples of XML for any product type are freely available upon request.

Overall the XML is structured rather simply. Few tags are required, XML tag ordering is not required, and unrecognized tags are ignored.

Data Formats

SPUD services returning meta-data generally return content of type XML, JSON or HTML. If, say, a standard web browser sets the HTTP header *Accept* value to "text/html" (which is normal), the service will attempt to return an HTML representation of the XML data. This is done for two reasons. The first is that many browsers (e.g. Safari) will not render XML at all. The second is that no browser supports a clickable link inside an XML document. Since there are many

instances where a clickable link is appropriate in a browser, e.g. in a query result returning a list of items.

Clients that can control their HTTP headers, e.g. wget, curl, can explicitly request XML or JSON. Most services that return Items, Thin representations of items or lists of the above will product the requested content type.

SPUD services also provide a method for overriding the output format when multiple formats are possible and the user wishes to access data in a particular format without utilizing the HTTP *Accept* header. The query parameter *output* can be set to one of [xml, html, json] (case insensitive) and this will set the *Accept* header of the request to **only** the indicated MIME type, e.g. "application/xml". Strings outside the above set or MIME types will be ignored. Services that return only a specific MIME type will return data in their specific format, ignoring the *Accept* header completely.

Individual Item Access

Individual item's metadata can be accessed by the GET HTTP method via URLs constructed as follows:
…/spudservice/PRODUCT_TYPE/ITEM_ID
where PRODUCT_TYPE must either be a specific product type, e.g. 'gmv', 'eventplot', 'earthmodel', etc. or generically, the string 'item' and ITEM_ID represents a single SPUD ID of the requested item. The 'item' designation will work for all product types. This has relevance when requesting multiple items, which may be of different product types.

Object Representation

When individual items are accessed via GET, the XML returned is a complete description of the item including all meta-data for the object **and** all attached objects with all their meta-data.

Also included are 'Link' tags that indicate the URLs of

- The item accessed
- Each attached object (if any) and its attached objects (if any)
- Links to Data Service URLS to access any data from data objects in native format
- Links to other URL: Primary References, External URLs, etc.

These Link items are valid clickable HTML links when the output is returned in HTML format.

Multiple Item (List) Access

Requesting multiple items is accomplished by using the HTTP GET method on URLs constructed as follows

```
http://www.iris.edu/spudservice/PRODUCT_TYPE?optional_query_parameters
```

As in the individual access case, the PRODUCT_TYPE string can be replaced by a specific product type or by the string 'item'. The latter allows querying across all product types.

Query parameters lists are constructed in standard URL format, e.g. item?evtminlat=20.0&evtmaxlat=40.0

The items returned can be limited to a specific product type by using that product type string in the URL.

The items returned can also be limited to those which meet query criteria that are specified via various query parameters (described below) that can be appended to the URL. These query parameters only have effect for queries accessing multiple item lists.

Subsets of data meeting the prescribed criteria may also be accessed using the query parameters *startitem* and *maxitems* described below.

Ordering is available based on Event (time, magnitude or depth), Product time or Submission time. Ordering can be ascending or descending.

Object Representation

When a list of items is accessed via GET, the XML returned is a single *Results* object that includes multiple XML representations of SPUD item within it. The individual item representations include a subset of the item's meta-data (the *thin* representation) and no attached objects or any other product specific meta-data.

Multiple Item Metadata Services

The count of the number of items that meet the specified criteria can be retrieved via URLs of this form.

`…/PRODUCT_TYPE/count?query_parameters`

A single integer is returned in text format.

The list of SPUD IDs can similarly be requested via

`…/PRODUCT_TYPE/ids?query-parameters`

A list of IDs, one per line, is returned

The *count* and *ids* URLs return their results in plain text. The *format* query parameter is ignored.

Sub-Item Object Access

Individual subcomponent objects of items (and lists of objects) can also be accessed directly. The returned meta-data is in XML format identical to the XML format of a standard item. The services used to update and create objects

associated with an item accept XML meta-data and raw data object data similarly to the services that perform the same functions for complete items.

Item Management Details

The creation, deletion and updating of items with the SPUD system is handled via the same URL structure outlined above. All SPUD services other than GET require HTTP authentication.

Creating and Updating Items

Creation and Updating of items in SPUD is handled via the HTTP methods POST and PUT respectively. SPUD accepts submissions in standard multipart/form format identical to standard web-based form submission.

Minimally, the POST and PUT services require that one of the objects submitted be a valid SPUD item in XML form which must be flagged as being in "application/xml" format. Data files associated with SPUD data objects can also be included in the submission.

The use of web-based and command line tools for submitting data to SPUD is beyond the scope of this document. Further information is available upon request.

Deleting Items

Deleting of items within the SPUD system is handled via URLs as described above.

5. Data Services

SPUD Data Services support streaming of native format data via the GET HTTP method. This allows clients to access the actual data stored in SPUD's various data objects, e.g. an MP4 movie attached to an Primary Data object.

SPUD Data Services support only the GET HTTP protocol and do not use a representational URL structure. All that is required to create a Data Service URL is the ID of the object and the type, i.e. Primary Data, Screen, etc.

Primary and Source Data Access

The data components of Primary and Source Data objects can be retrieved via URLs of this form:

```
…/data/id
```

Where id is the ID of the requested object.

Source data for a particular item can also be requested through the URL:

`…/PRODUCT_TYPE/id/source`

The data is streamed to the client with content type (MIME type) set appropriately.

File Data

The data components of File Data objects can be retrieved via URLs of this form:

`…/filedata/id`

Where id is the ID of the requested object.

The data is streamed to the client with content type (MIME type) set appropriately.

Screen and Thumbnail Data

The data components of Screen and Thumbnail Data objects can be retrieved via URLs of this form:

`…/data/alternate/id`

Where id is the ID of the requested object.

The data is streamed to the client with content type (MIME type) set appropriately.

Byte Ranging

SPUD data services support byte ranging to support clients requiring this feature. A discussion of this is beyond the scope of this document. File Data services do not currently support byte-ranging.

Data Access Via Media Keys

Primary and File Data objects may include meta-data that is used by SPUD in the display and positioning of various media. These meta-data are known as ***GroupKey*** and ***OrderKey*** . For instance, the standard vertical component GMV movie of a GMV item has Group Key = 1. Images in Event Plot items are grouped into image galleries within SPUD via GroupKey. OrderKey controls their order within those galleries.

Individual Media Access

Services are available to target individual media data for SPUD items based on GroupKey and OrderKey. As neither GroupKey nor OrderKey are required meta-data and are not required to be unique to media objects, it is possible to have an item contain a list of media objects that, say, possess no GroupKey or

OrderKey values or contain multiple objects with the same GroupKey. Given that a specification of GroupKey and / or OrderKey (or the lack of either or both) may reference multiple data objects, an ***Index*** parameter is also included to allow selection of individual media objects from the list constrained by any GroupKey or OrderKey parameters. If no index parameter is specified the first matching object's data is returned.

Primary Data objects are available via URLs of this type:

…/PRODUCT_TYPE/ID/media[optional_query_parameters]

File Data objects are available via URLs of this type:

…/PRODUCT_TYPE/ID/filemedia[optional_query_parameters]

As it may be useful to know the number of media objects meeting any criteria, the counts are available at these URLs

…/PRODUCT_TYPE/ID/mediacount[optional_query_parameters]

…/PRODUCT_TYPE/ID/filemediacount[optional_query_parameters]

Parameter	Use
group	Requires that the object returned have GroupKey equal to the provided integer value.
order	Requires that the object returned have OrderKey equal to the provided integer value.
index	Return the object at this position in the list of all media objects, possibly constrained by the group and / or order parameters.

Multiple Item Multiple Media Access

Services are available to collectively group and return media data for multiple SPUD items based on GroupKey and OrderKey pairs.

The collection of items for which media is to be returned are based on the same set of query parameters as for Multiple Item List queries above.

In addition there is a parameter which allows the client to request a subset of the media available to those matching a list of GroupKey / OrderKey pairs. This query parmater, which is only available for this service is *mediakey*. It's value should be a set of one or more pairs of group and order key integers, separated by a ',' character. Each pair is separated by a ':' character.

E.g. …/PRODUCT_TYPE/media?mediakey=2:1,2:2

Will select media with group key 2, order key 1 and group key 2, order key 2 from all active items of the selected product type.

The generic product type 'item' is available for this service.

If no order key value is given or the ':' character is missing, all matching media with any order key are returned.

If no group key is given (this implies the presence of the ':' character), then all matching media with any group key are returned.

If no group key and / or order key are specified, all media from all matching items is returned.

This service can often require significant time to respond for certain queries and this can easily cause the HTTP connection to 'time out'.

Parameter	Format
mediakey	KEY_PAIR1, KEY_PAIR2, … KEY_PAIR = groupkey:orderkey= | groupkey | :orderkey

6. Other Services

GMV Media Shortcuts

Shortcut URLs are provided for accessing vertical component and three component GMV visualization media. These provide direct streaming access to the media associated with a particular GMV. Note: not all GMV items possess 3 component media.

…/gmv/ID/vertical

…/gmv/ID/3C

Bundling

Bundling is the term used for the creation of a compressed ZIP file containing 1 or more SPUD items' meta-data and media. The ZIP file returned by this service contains a separate subdirectory for each item requested. Each items' subdirectory contains its meta-data file (XML) and all its Primary and Attachment data. No alternate data (Thumbnail, Screen) or original source data is included.

The bundling service is available as a standard HTTP GET service using the following URL

…/bundle?id1&id2…

where a list of SPUD Ids is supplied as query parameters without values, i.e. no "=value". Unknown ids are ignored.

The bundling service is also available via an HTTP POST service at the …/bundle URL. It requires a multipart/form-data payload which contains a single line referenced as "id", containing a comma separated list of SPUD IDs.

The size of a bundle in bytes (before ZIP compression) can be accessed via a URL of this form:

…/bundle/size?id1&id2…

Extracting

Extracting is similar to bundling, but returns all data associated with an item and strives to preserve any original file structure to the submitted data. Substitute the string "extract" for "bundle" in the above URLs.

Moment Tensor Services

Retrieving a Moment Tensor Item via Event Name

Valid Moment Tensor items in SPUD contain the GCMT Event Name from the original NDK data used in creation of the item. While the Event Name is not unique among SPUD items of Moment Tensor product type, there is only one *active* item within the SPUD system with this Event Name. Therefore, certain services are provided by which active Moment Tensor items can be selected via the GCMT Event Name.

To retrieve the active Moment Tensor item corresponding to a GCMT Event Name use this URL:

…/momenttensor/gcmtid/EV_NAME

where EV_NAME is the GCMT Event Name string.

Retrieving a List of Event Names

A list of distinct Moment Tensor event names based on standard item criteria can be retrieved vis the URL

…/momenttensor/gcmtids[?optional_query_parameters]

Source NDK

The original ASCII NDK used to generate Moment Tensor products in SPUD can be accessed via URLs in either of the two forms below

…/momenttensor/ID/source

…/momenttensor/ID/ndk

One can also access the NDK for an active Moment Tensor item though its GCMT Event Name

…/momenttensor/gcmtid/EV_NAME/ndk

Moment Tensor Focal Mechanism Images

A focal mechanism image in PNG format for a particular Moment Tensor item is available at the following URL

…/momenttensor/ID/focalmechanism

One can also access the focal mechanism image for an active Moment Tensor item though its GCMT Event Name

…/momenttensor/gcmtid/EV_NAME/focalmechanism

The following table lists the available query parameters for this service

Parameter	Use
size	Length and width of the returned image in pixels. Default: 200
linewidth	Weight (in pixels) of the focal mechanism borders. Default: 2
red	Red level of focal mechanism image. Range: 0-1. Default: 0.8
green	Green level of focal mechanism image. Range: 0-1. Default: 0.1
blue	Blue level of focal mechanism image. Range: 0-1. Default: 0.1
ticks	Display image with compass points. Valid values: Y|y|N|n. Default: n

Moment Tensor QuakeML

A QuakeML representation of a SPUD Moment Tensor item can be retrieved via a URL of this form:

…/momenttensor/ID/quakeml

Alternatively, the QuakeML representation can be retrieved via a URL based on the GCMTeventID

…/momenttensor/GCMTID/quakeml

The usual formatting options apply, i.e. format=xml | html

The QuakeML representation is provided via the QuakePy package available from ETHZ at www.quake.ethz.ch/quakepy

7. Reference

Available SPUD Product Types

The table below lists the currently SPUD product types and their URL strings active in the SPUD system as of September 2011.

Product Type	URL String
Ground Motion Visualizations (GMV)	gmv
Event Plots	eventplot
GCMT Moment Tensors	momenttensor
Back Projections	backprojection
Electromagnetic Transfer Functions	emtf
IDA Calibrations	calibration
NEIC Event Bulletins	eventbulletin
WWSSN Film Chips	filmchip
ANF Station Digests	stationdigest
Earth Models	earthmodel

Query Parameters

Query parameters are used exclusively on GET services that return lists, counts and ids of items. These query parameters have no effect on any other services

Generic Item Parameters

These parameters apply to all product types.

Base Parameters

Parameter	Valid Values	Use
startitem	Integer >= 0	First item of list to return
maxitems	Integer > 0	Maximum number of items to return
active	YN*	Return active, inactive items or either Default: 'Y'
subtype	STRWILD	Items matching specified subtype [wildcards allowed]
productid	STRWILD	Items matching specified productid [wildcards allows]
tags	STRWILD	Item matching specified tags.
hasevent	YN	Return only items which are associated with an event
format	FORMAT	Selects either XML or HTML output format for meta-data

Results Ordering Parameters

Parameter	Valid Values	Use
sort	SORT	Orders results by event, submit time, etc.
evtsort	EVTSORT	Orders results by event mag, depth, etc. (applies only if sort=event. See above)
order	ORDER	Ascending or descending order

Event Based Parameters

These parameters allow the user to limit the query results to items possessing events whose latitude, longitude, depth and date fall within the specified bounding limits specified by the *min* and *max* parameters below.

Specification of these parameters will limit queries to those items possessing event information.

Parameter	Valid Values	Use
evtminlat	LATRANGE	Minimum latitude of event bounds
evtmaxlat	LATRANGE	Maximum latitude of event bounds
evtminlon	LONRANGE	Minimum longitude of event bounds
evtmaxlon	LONRANGE	Maximum longitude of event bounds
evtmindepth	FLOAT	Minimum depth of event bounds§
evtmaxdepth	FLOAT	Maximum depth of event bounds§
evtminmag	FLOAT	Minimum magnitude of event
evtmaxmag	FLOAT	Maximum magnitude of event
evtstartdate	DATE	Earliest date of event
evtenddate	DATE	Latest date of event
hasevent	BOOLEAN	Return only items which have an event
eventid	Integer > 0	ID of associated prime event (if any)
gcmtid	STRWILD	ID of GCMT event (if any)

Product Location Based Parameters

These parameters limite the query results to those items whose product region, depth range, and time span **overlap** (inclusively) those of the specified query parameters.

§ Depths can be negative to express elevations.

Parameter	Valid Values	Use
prodminlat	LATRANGE	Minimum latitude of product region
prodmaxlat	LATRANGE	Maximum latitude of product region
prodminlon	LONRANGE	Minimum longitude of product region
prodmaxlon	LONRANGE	Maximum longitude of product region
prodmindepth	FLOAT	Minimum depth of product region§
prodmaxdepth	FLOAT	Maximum depth of product region§
prodstartdate	DATE	Earliest date of product relevancy
prodenddate	DATE	Latest date of product relevancy

Seismic Station Based Parameters

Parameter	Valid Values	Use
station	STRWILD	Station name
network	STRWILD	Network name
channel	STRWILD	Channel name
location	STRWILD	Location name

Provenance Based Parameters

Parameter	Valid Values	Use
createstartdate	DATE	Earliest product creation date
createenddate	DATE	Latest product creation date
submitstartdate	DATE	Earliest product submission date

submitenddate	DATE	Latest product submission date
modstartdate	DATE	Earliest product modification date
modenddate	DATE	Latest product modification date
submittername	STRWILD	Submitter name
submitterorg	STRWILD	Submitter organization
creatorname	STRWILD	Creator name
creatororg	STRWILD	Creator organization
creatingapp	STRWILD	Creating application

EM_TF Product Specific Parameters

Depth values can be negative to represent elevation

Parameter	Valid Values	Use
project	STRWILD	Project
survey	STRWILD	Survey
releasestatus	STRWILD	Release Status
remoteref	STRWILD	Remote Reference Name
sitename	STRWILD	Site Name
remotesite	STRWILD	Remote Site Name
minquality	0 <= Integer <= 5	Minimum Product Quality

Event Bulletin Product Specific Parameters

Depth values can be negative to represent elevation

Parameter	Valid Values	Use
contributor	STRWILD	Contributor

Earth Model Product Specific Parameters

Parameter	Valid Values	Use
modelkeyword	STRWILD	Model / product ID keyword

Parameter Format Key

Value	Definition
STRWILD	String allowing wildcard characters: [?*][2]
STRWILD	String allowing wildcard characters: [?*]
YN	Y | N | y | n
YN*	Y | N | y | n | * ('*' indicates either state)
LATRANGE	Decimal float [-90.0 – 90.0]
LONRANGE	Decimal float [-180.0 – 180.0]
FLOAT	Float in decimal or exponential notation, e.g. -20.23, -2.023e1
DATE	Date in format: yyyy-mm-ddThh:mm:ss in 24 hour format
SORT	submittime | event | time

[2] '?' matches a single character. '*' matches zero or more of any character. Similar to UNIX glob. '\' is the escape character, e.g. "*" matches a single '*' character.

EVTSORT	mag | depth | time

8. Appendix

HTTP Status Responses

SPUD Services return standard HTTP status codes in the HTTP response header that depend on the result of the request. Below is a list of commonly used codes and their meanings.
Additionally, SPUD Services may return a plain text string in the message body indicating the type of error encountered.

200 OK

The request has succeeded. This is the response to a successful GET, PUT or DELETE request

201 Created

A new item has been created. This is the response to a successful POST operation. The response header to a successful POST request will contain the SPUD Service URL of the newly created item in the *Location* field.

206 Partial Content

This is the response to a successful request for a byte ranging request for data. This is only supported for SPUD Data Services.

400 Bad Request

This is the response to an unsuccessful request, generally due to malformed or erroneous query parameters.

401 Unauthorized

This is the response to an unsuccessful request to a service that requires authentication, generally a POST, PUT or DELETE service (all of which require authentication).

404 Not Found

This is the response to a request that can't be services. In general, this is due to an incorrect URL.

500 Internal Service Error

An error has occurred during the processing of the service call.

9. Examples

How can I use SPUD Services from the Command Line?

HTTP requests can be made using the UNIX command line tools **wget** and **curl**. These are available for a variety of other operating systems, including Mac OSX and Windows. E.g.

```
wget -Ogmv_ids.txt www.iris.edu/spudservice/gmv/ids
```

will retrieve a list of SPUD IDs, one per line, for all active GMVs in the SPUD system. The result will be stored in the file named gmv_ids.txt

What can I do with this big list of IDs that SPUD services returned?

Use the UNIX command line tool xargs.

If the list of IDs is from a SPUD Service that returns IDs one per line, you can use UNIX command line tools such as wget and xargs command to make repetitive calls to SPUD services. For example: to access each NDK file associated with a list of IDs extracted from SPUD services all in one blow, try…

```
wget -qO- www.iris.edu/spudservice/momenttensor/ids?evtstartdate=2004-12-26T00:00:00&evtenddate=2004-12-27T00:00:00&evtminmag=7.0 | xargs -n 1 -I ID wget -O ID.ndk www.iris.edu/spudservice/momenttensor/ID/ndk
```

The "-I" option of xargs will substitute the output of the first command (A list of IDs) into the 2nd wget command wherever the string 'ID' appears. Hence, the result will be a number of ndk files each created by one call to the ndk service.

Moment Tensors

How do I get a focal mechanism 'beachball' for a Moment Tensor with SPUD ID 1234?

…/spudservice/momenttensor/1234/focalmechanism

How do I get a list of IDs for all Moment Tensor items associated with events with magnitudes greater than 7.0 on 27/12/2004?

…/spudservice/momenttensor/ids?evtstartdate=2004-12-26T00:00:00&evtenddate=2004-12-27T00:00:00&evtminmag=7.0

How do I get the PNG 'beachball' files associated with that list of IDs?

Almost exactly the same way as getting the ndk [described above]. Just use the focalmechanism URL rather than the ndk URL.

```
wget -qO- www.iris.edu/spudservice/momenttensor/ids?evtstartdate=2004-12-26T00:00:00&evtenddate=2004-12-27T00:00:00&evtminmag=7.0 | xargs -n 1 -IID wget -O ID.png www.iris.edu/spudservice/momenttensor/ID/focalmechanism
```

The "-I" option of xargs will substitute the output of the first