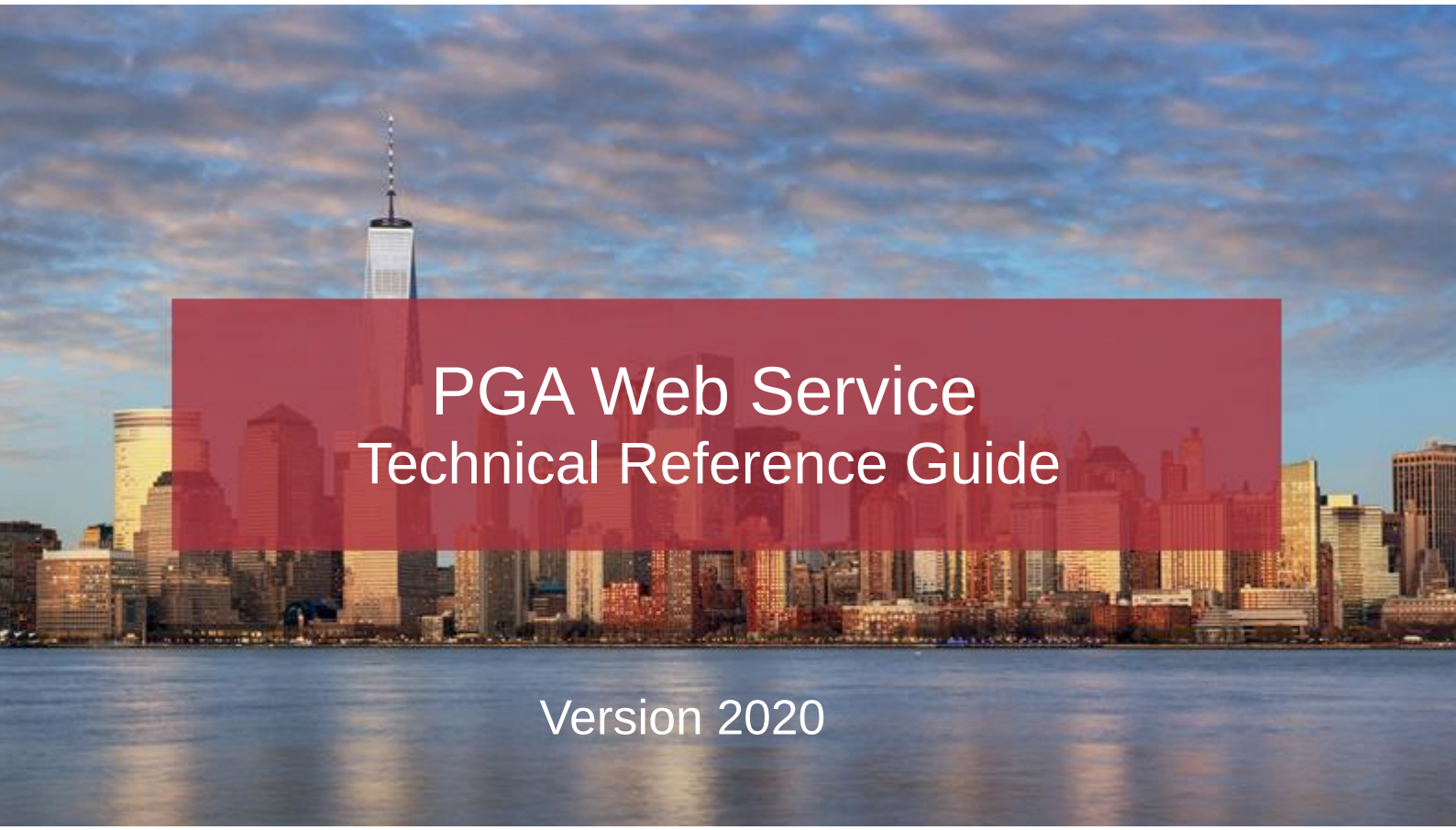


PGA Web Service
Technical Reference Guide

Version 2020

Confidentiality, Copyright Notice & Disclaimer

Due to a policy of continuous product development and refinement, TEOCO Ltd. (and its affiliates, together “TEOCO”) reserves the right to alter the specifications, representation, descriptions and all other matters outlined in this publication without prior notice. No part of this document, taken as a whole or separately, shall be deemed to be part of any contract for a product or commitment of any kind. Furthermore, this document is provided “As Is” and without any warranty.

This document is the property of TEOCO, which owns the sole and full rights including copyright. TEOCO retains the sole property rights to all information contained in this document, and without the written consent of TEOCO given by contract or otherwise in writing, the document must not be copied, reprinted or reproduced in any manner or form, nor transmitted in any form or by any means: electronic, mechanical, magnetic or otherwise, either wholly or in part.

The information herein is designated highly confidential and is subject to all restrictions in any law regarding such matters and the relevant confidentiality and non-disclosure clauses or agreements issued with TEOCO prior to or after the disclosure. All the information in this document is to be safeguarded and all steps must be taken to prevent it from being disclosed to any person or entity other than the direct entity that received it directly from TEOCO.

TEOCO and Netrac® are trademarks of TEOCO.

All other company, brand or product names are trademarks or service marks of their respective holders.

This is a legal notice and may not be removed or altered in any way.

Your feedback is important to us: The TEOCO Documentation team takes many measures in order to ensure that our work is of the highest quality.

If you found errors or feel that information is missing, please send your Documentation-related feedback to Documentation@teoco.com

Thank you,

The TEOCO Documentation team

Change History

This table shows the change history of this guide:

Edition	Date	Reason
1	31 March 2020	First edition.

Table of Contents

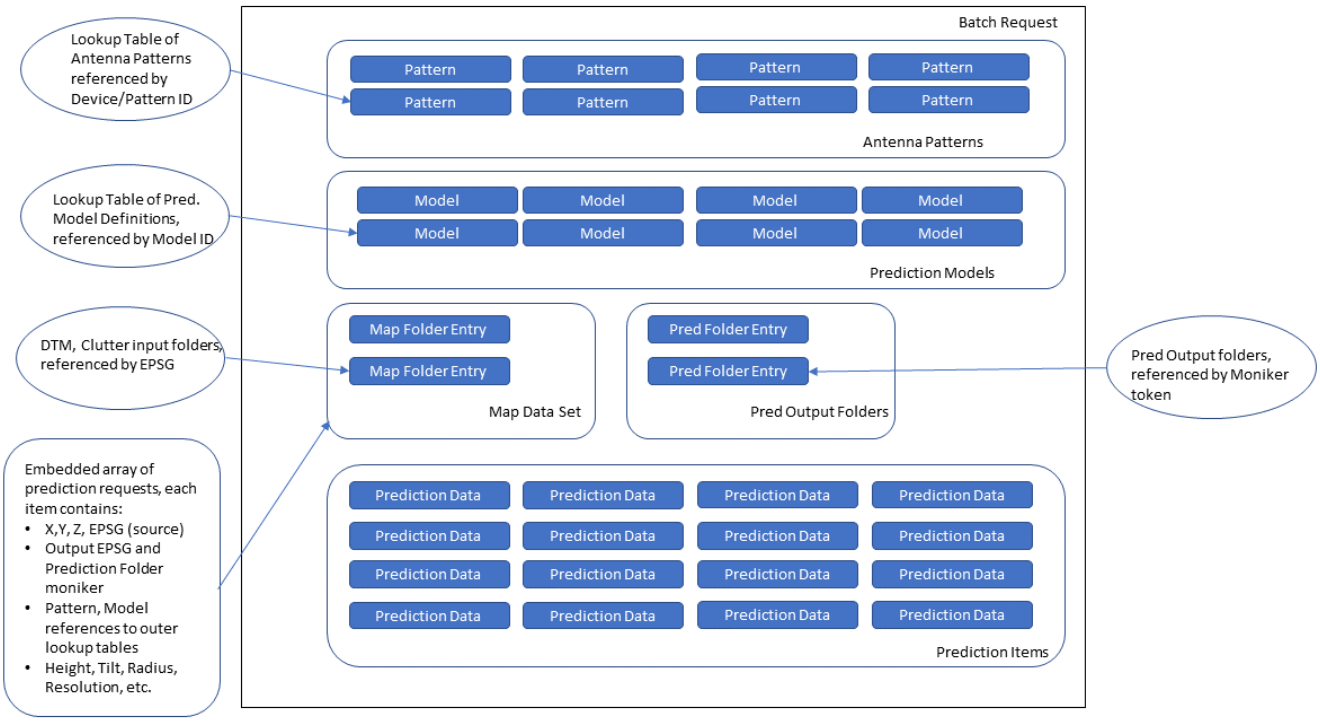
1 Introduction

The **Predictions Generation API** (**PGA**) is a web service which is layered over TEOCO's ENTERPRISE prediction engine, and is designed to allow a system or an end user to request the generation of predictions.

The service accepts work in the form of batched requests, where a batch contains all the information necessary to generate predictions:

- The (cellular) antennas, propagation models and map data that you want to make available for use in each job
- The folder in which to store the generated predictions
- One or more prediction jobs, which define:
  - The location for which you want to predict
  - The specific map data and output folder you want to use
  - The specific antenna you want to use and its instance parameters
  - The specific propagation model you want to use and its instance parameters

This picture shows the composite structure of a batch request:



Batch request structure

The service supports concurrent requests from multiple callers. It will submit a batch request onto a queue and return its token immediately. The batch request will be associated with a number of enumerable states that can be interrogated by the client, such as: Pending, Processing, Completed, Failed, Cancelled.

The service has the ability to provide information that can be fed into the Prediction Access Module (PAM).

Note: The PGA service does not currently support a RESTful interface.

2 Installing the PGA

To install the PGA:

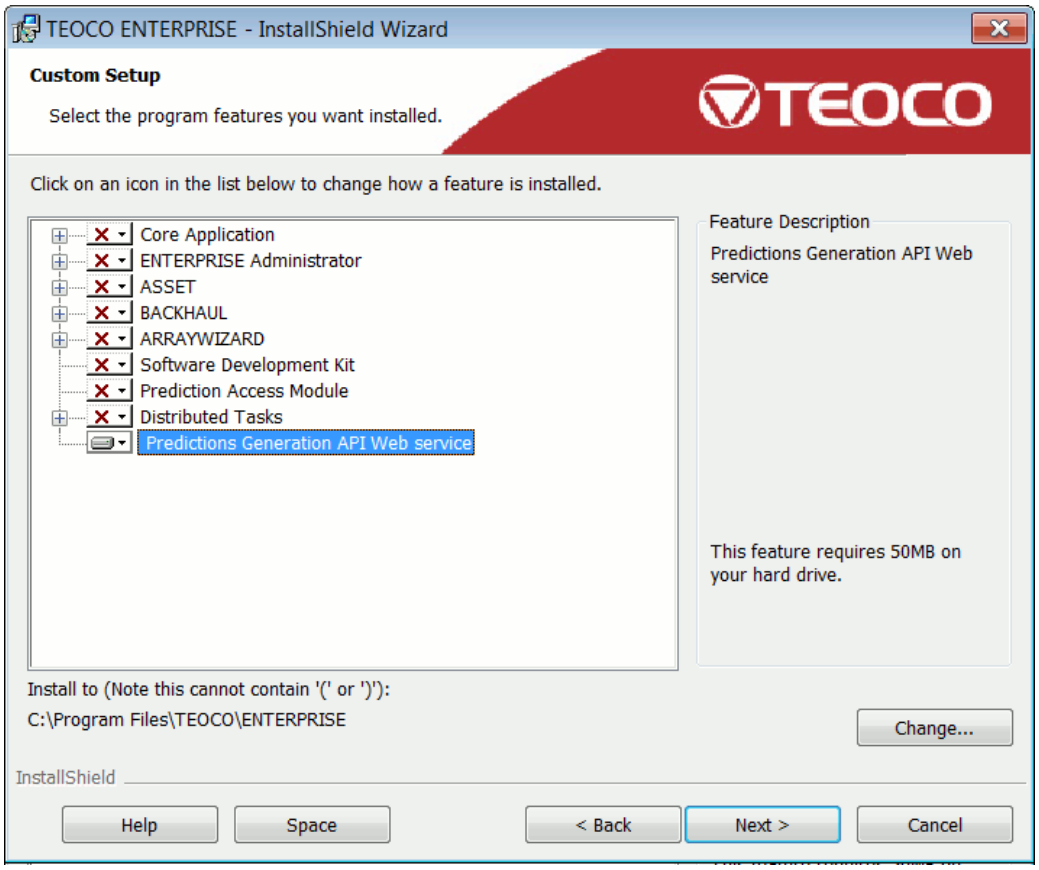
1. The PGA installation is part of the normal ENTERPRISE installation. Please refer to the *ENTERPRISE Installation and Administration Guide* for full information on this.

2. Perform the installation as directed, but when you reach the **Setup Type** page, ensure that you select '**Custom**'. This will enable you to select which products and options to install.

   Click **Next**.

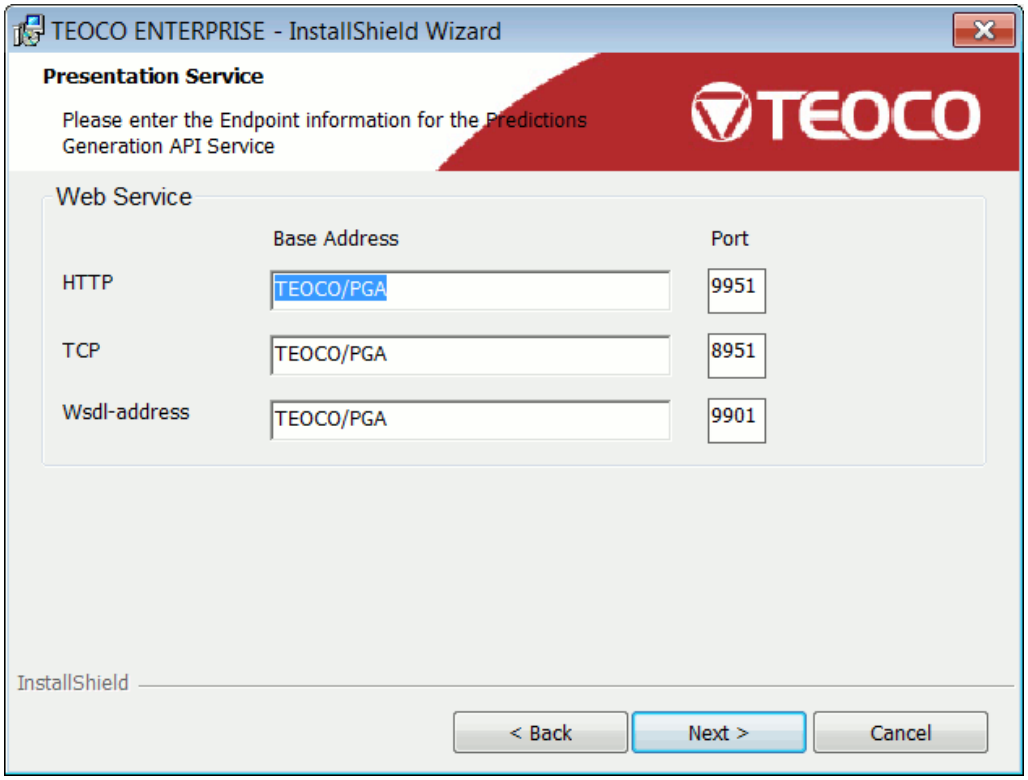
3. Select the **'Predictions Generation API Web service'** option.

   The other options you select will depend on your precise requirements. If you only want to install PGA, there is no need to select any of the other options. This picture shows an example:



4. Click **Next**.

The **Prediction Service** screen appears, as shown here:



If necessary, you can amend the endpoint information for the Web service, but it is generally advisable to use the defaults.

Notes:

- Ensure you have different port numbers for each endpoint, otherwise the installation will fail.
- The installer will attempt to add an exception to the installed firewall (if present) to allow the application to be contacted externally from the machine hosting it.

Click **Next**.

5. On the **License Server** page, specify the name of the license server.

   Click **Next**.

6. If you want to use the service as a client of an existing Prediction Distribution system, please enter the valid connection details as described in Configuring the Distribution Coordinator Location in Section 2.1.

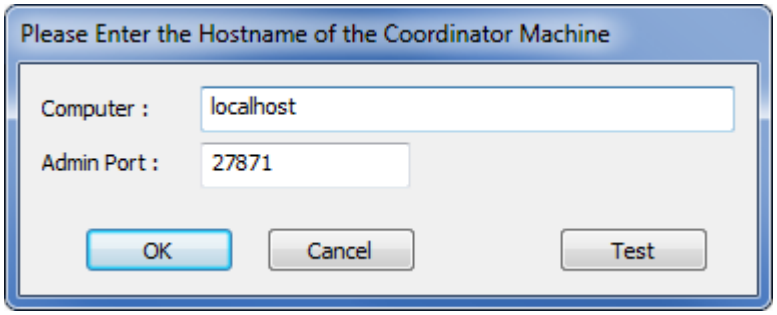
   Otherwise, click **Next**.

7. Complete the subsequent steps of the installation, as described in the *ENTERPRISE Installation and Administration Guide*.

2.1 Configuring the Distribution Coordinator Location

If you intend to use the service as a client of an existing Prediction Distribution system:

1. Please enter valid connection details. Here is an example:



2. Click **Test** to confirm the connection.
3. Click **OK** if successful.

- or -

If distribution is not required at this time, click **Cancel** to complete the installation.

Important: If distribution will be required at some stage after the installation, this must be configured *before* enabling distribution on the service (which is described in Configuring Operational Parameters in Section 3.2).

To do this, from the Windows **Start** menu, point to **All Programs**, **TEOCO**, **ENTERPRISE**, **Task Distribution**, and then click **Set Coordinator Hostname**.

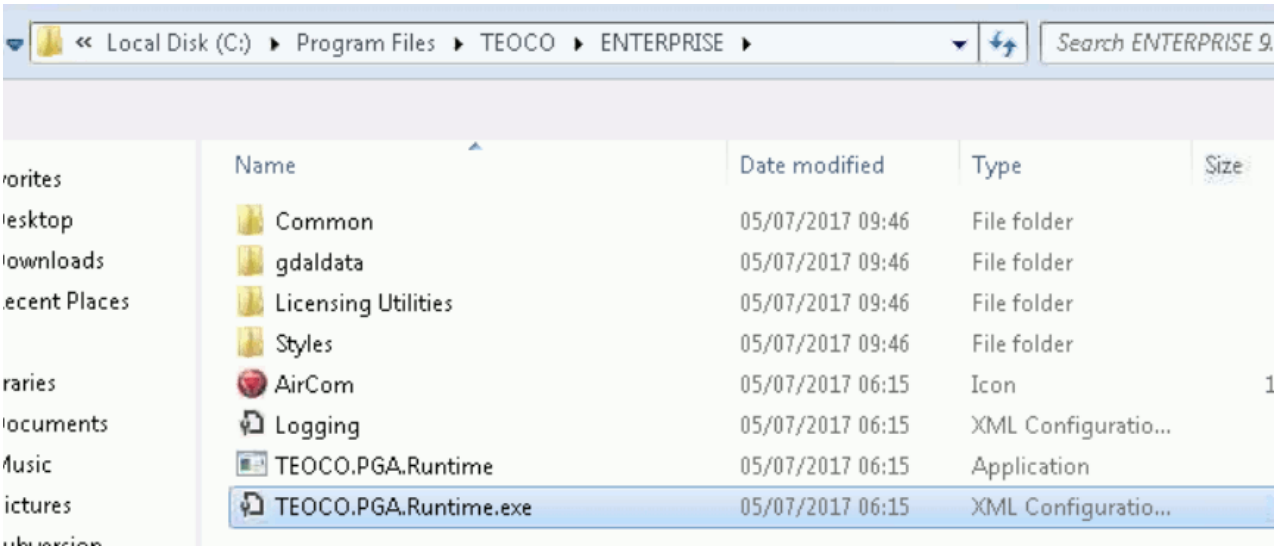
---

3 Configuring PGA

Before using PGA, you can perform a number of configuration tasks:

- Define the endpoint addresses used by the service
- Define a number of behavioural settings, based around:
    - The clean-up process
    - Restrictions that protect the service from being overloaded, or impacting the host machine
    - How PGA interacts with ASSET's distribution system

These configuration options are defined in the 'TEOCO.PGA.Runtime.exe.config' file, which is located in the installation folder (for example, C:\Program Files\TEOCO\Enterprise):



This picture shows an example file:

3.1 Configuring Service Endpoints

The ‘address-configuration’ section enables you to configure the endpoint addresses used by this service:

```
<address-configuration address="localhost">
 <services>
  <service name="TEOCO.PGA.Service.PGAServiceImpl" address=""
      wsdl-address="TEOCO/PGA" wsdl-port="9901" wsdl-scheme="http"
      majorVersion="0"  minorVersion="1" >
   <endpoints>
      <endpoint name="TCP" address="TEOCO/PGA" ignorePort="override" Port="8951" />
      <endpoint name="HTTP" address="TEOCO/PGA" ignorePort="override" Port="9951"/>
   </endpoints>
  </service>
 </services>
</address-configuration>
```

The ‘web-api’ section enables you to configure the REST endpoint port and the api-key used by this service:

```
<web-api port="8683">
  <api-key key="teoco">
    <role name="Create" />
    <role name="Update" />
    <role name="Delete" />
  </api-key>
</web-api>
```

The api-key is used to secure some of the functionality of the REST endpoint. It may be any valid string, without spaces or reserved characters. The corresponding PGA roles that are permitted for that api-key are nested beneath it.

In the below example you can see the default “teoco” key and then two more named “demo1-key” and “demo2-key” that can be used to control the access to Create and Update/Delete roles independently:

```
<web-api port="8683">
  <api-key key="teoco">
    <role name="Create" />
    <role name="Update" />
    <role name="Delete" />
  </api-key>
  <api-key key="demo1-key">
    <role name="Create" />
  </api-key>
  <api-key key="demo2-key">
    <role name="Update" />
    <role name="Delete" />
  </api-key>
</web-api>
```

The “teoco” api-key is provided by default as an example, and should be removed or replaced with a suitably secure key in production.

Note: The PGA must be restarted for any configuration changes to take effect.

3.1.1 Web Service Interface

This section deals with the service contract and objects that make up the primary interface for communicating with the service.

The base endpoint address for this interface (WS-HTTP) is:

Error! Hyperlink reference not valid.

A Microsoft-compatible “netTCP” endpoint is:

netTCP://{server}:8951/TEOCO/PGA/TCP

Its WSDL can be retrieved from this URI:

Error! Hyperlink reference not valid.

Its REST API and Swagger UI endpoint is:

http://{server}:8683/

3.2 Configuring Operational Parameters

The ‘pga-configuration’ section enables you to configure behavioural settings:

```
  <!--This section is used to configure pga specific behaviour settings -->
  <pga-configuration>
    <job-tracker-cleanup interval-mins="5" delta-mins="60" />
    <limits batch-count="1000" jobs-per-batch-count="1000" default-meta-data-item-
count="1000" />
    <distribution distribution-enabled="false" job-priority="Medium" />  <!-- job-
priority can be 'Lowest'; 'Low'; 'Medium'; 'High'; 'Highest' or a custom numeric value
string (currently 1..10000) -->
    <pred-folder-config-defaults max-storage-before-cleanup-gigabytes="1" cleanup-free-
space-target-percent="20" cleanup-check-interval-mins="15" max-file-write-retries="10"
/>  <!-- default values that take effect if the 'config.dat' file is missing in the
target pred-folder-->
  </pga-configuration>
```

You can define the following parameter categories:

- Job-Tracker-Cleanup: Enables you to define the automatic clean-up process. When a batch reaches one of the completion states (‘Completed’, ‘Cancelled’ or ‘Failed’), its status within the PGA is updated and the collected metadata will be retained for a fixed amount of time.

  It is the responsibility of the PGA client to retrieve this metadata before an automatic clean-up process removes this data to save memory. The parameters in this category determine this behaviour, by specifying how regularly the automatic clean-up process runs and how old the metadata can be before it is removed.

- Limits: Enables you to prevent the service getting overloaded, or from adversely affecting the host machine.

- Distribution: Enables PGA to interface with the existing ASSET Distributed Prediction system.

  If the service is not configured to enable distribution, all predictions are performed locally.

- Pred-Folder-Config-Defaults: Enables you to specify the maximum amount of space that predictions can occupy before automated cleanup is allowed to run; the percentage of allowed maximum space that should be available after cleanup completes; the time interval at which cleanup checks occur; the maximum number of re-tries the system is allowed to make.

  Note: The values under the Pred-Folder-Config-Defaults category are defaults in the sense they only take effect if the ‘Config.dat’ pred-folder configuration file is missing from the target pred-folder when PGA processes the first job using a given pre-folder. Also, these defaults will be persisted in a newly written ‘Config.dat’ file.

For a more detailed description of these parameter categories, see the table on the next page.

This table describes the parameters in each category (to be read in conjunction with the previous page):

Category	Parameter	Description	Default
job-tracker-cleanup	interval-mins	The interval (in mins) with which the clean up mechanism repeats. For example, if the interval is 5 minutes, then the mechanism will scan the list every 5 minutes, looking for any batches outside the window determined by the 'windows-mins' parameter.	5
	delta-mins	The age (in mins) of a batch before it is considered old enough to be removed from the list by the clean up mechanism, assuming the batch is in one of the completion states.	60
limits	batch-count	The maximum number of batches the service is allowed to work or hold data on at any one time.	1000
	jobs-per-batch-count	The maximum number of jobs on any one batch that the service is allowed to accept.	1000
	default-meta-data-item-count	The 'FetchMetaData' method call (described in PGA Web Service Description in Section 4) looks at each job in a specified batch (respecting the pagination parameters if specified), calculates the meta data appropriate for each job, and then returns it to the caller. Because the size of the metadata XML greatly exceeds the equivalent job XML, a lower limit is required to prevent too many being returned in one call, thus exceeding the maximum allowed size of a message.	1000
distribution	distribution-enabled	Indicates whether: The service performs the jobs locally ('False'). - or - Connects to a configured distribution coordinator and hands off all prediction job requests to that system instead ('True). **Warning**: You must ensure you have already configured the Distributed Coordinator location, otherwise the jobs will be performed locally. See Configuring the Distribution Coordinator Location in Section 2.1.	False
	job-priority	Determines the priority given to the jobs passed to the distribution system, in relation to its other clients. It has pre-defined values of Lowest, Low, Medium, High, Highest, but also supports a numeric value between 1 (Lowest) and 10,000 (Highest)	Medium

Category	Parameter	Description	Default
pred-folder-config-defaults	max-storage-before-cleanup-gigabytes	The maximum amount of space (in gigabytes) that predictions can occupy before automated cleanup is allowed to run (this is the value that is normally derived from the '[Max Disk Space]' setting in the 'Config.dat' pred-folder configuration file).	1
	cleanup-free-space-target-percent	The percentage of allowed maximum space that should be available after automated cleanup completes (this is the value that is normally derived from the '[Percent Free Space Target]' setting in the 'Config.dat' pred-folder configuration file).	20
	cleanup-check-interval-mins	The time interval (in minutes) at which automated cleanup checks occur (this is the value that is normally derived from the '[Cleanup Interval Time (Mins)]' setting in the 'Config.dat' pred-folder configuration file).	15
	max-file-write-retries	The maximum number of re-tries the system is allowed to make following failure to complete a prediction write operation (this is the value that is normally derived from the '[File Write Retry Count]' setting in the 'Config.dat' pred-folder configuration file).	10

3.3 Configuring PGA Eventing

PGA is now able to publish job status updates and job completion events via RabbitMQ message bus. The 'pga-event-configuration' section of the configuration file is used to enable and configure this feature:

```
<pga-event-configuration msgid="teoco.events.asset" msgver="1.0" msghlp=""
     enabled="false">
  <drivers>
    <driver name="rabbit"
     type="TEOCO.EWS.Framework.Messaging.RabbitMQ.MessageConfigSection,
     TEOCO.EWS.Framework.Messaging" />
  </drivers>
  <publishers>
    <rabbit name="PGA" Provider="amqp:" RequestQueue="topic://PGA/PGAQueue"
     Broker="localhost" AutoCreateQueues="true" TimeToLive="3600"
     RoutingKey="PGAEvent"/>
  </publishers>
</pga-event-configuration>
```

The “pga-event-configuration” section contains several important configuration settings:

Parameter	Default	Description
Enabled	false	Enables or disables the Eventing feature
msg-id	teoco.events.asset	Used to recognise PGA events
msgver	1.0	Provides simple version control management
Msghlp		Customisable help string

The “rabbit” section within the “publishers” contains the configuration for RabbitMQ broker, exchange and queues:

Parameter	Default	Description
Broker	localhost	Server where the RabbitMQ is running.
name	PGA	Name of the Exchange to create and use within RabbitMQ
Provider	ampq:	This is the provider prefix for RabbitMQ, and should not be changed.
RequestQueue	topic://PGA/PGAQueue	URL to the queue. Format is <type>://<exchange>/<queue name>
RoutingKey	PGAEvent	Routing key used for the queue.
AutoCreateQueues	true	Determines if the queue is auto-created if it doesn’t exist.
TimeToLive	3600	Number of seconds the published messages will be held in the queue without being consumed, before it is timed out and discarded.

Multiple “rabbit” sections can be added within the “publishers” section, if PGA is required to publish the events to multiple exchanges or to different RabbitMQ instances. In the below example, all PGA events will be published to both ServerOne and ServerTwo.

```
<publishers>
    <rabbit name="PGA" Provider="amqp:" RequestQueue="topic://PGA/PGAQueue"
           Broker="ServerOne" AutoCreateQueues="true" TimeToLive="3600"
           RoutingKey="PGAEvent"/>
    <rabbit name="PGA" Provider="amqp:" RequestQueue="topic://PGA/PGAQueue"
           Broker="ServerTwo" AutoCreateQueues="true" TimeToLive="3600"
           RoutingKey="PGAEvent"/>
</publishers>
```

4 PGA Web Service Description

This section describes the service contract that can be used to communicate with the service:

- Contract implementation
- Common data types
- Submit method

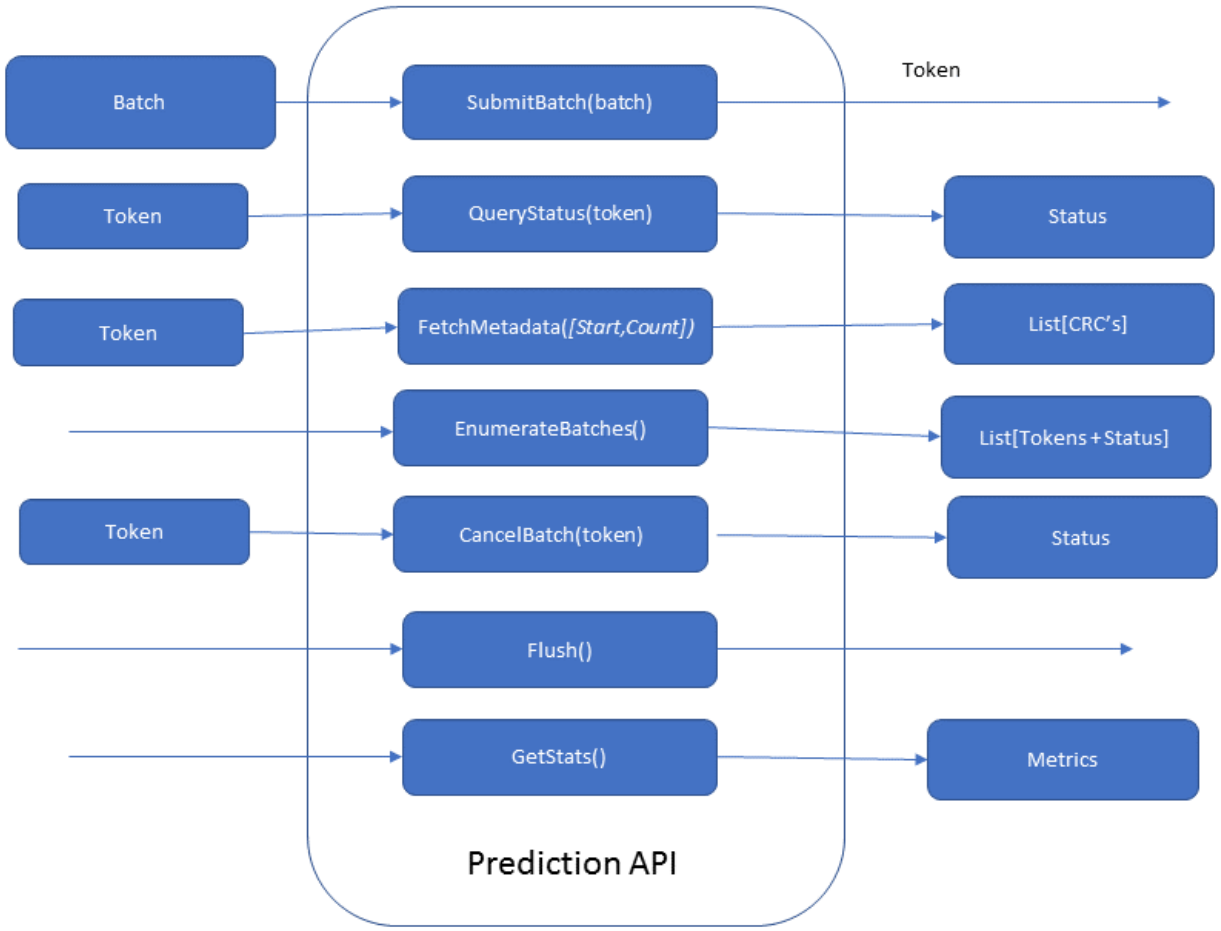
This table gives an overview of the available methods:

Method	Input	Output	Description
Submit	Batch object	Token ID	Enables a caller to request a batched number of predictions from a single request. The returned token ID allows the caller to interrogate the progress of the job. See Submit Method in Section 4.4.
Query	Token ID	Status object	Enables a caller to interrogate the status of a previously submitted job. The status object will summarise whether the job is pending, processing, completed and so on. Any progress completion of the batch (for example, 17 of 1500) will be available here alongside any errors encountered from the system. See Query Method in Section 4.5.
EnumerateBatches	None	Job status summary	Enables a caller to list all jobs present on the API, their current status (pending, processing and so on) and their token ID. See EnumerateBatches Method in Section 4.6.
EnumerateModels	None		Enables a caller to list all propagation models available to be used by the service. See EnumerateModels Method in Section 4.7.
Cancel	Token ID	Status object	Enables a caller to cancel a job. A summary of its progress (if any) will be made available via the Status object. See Cancel Method in Section 4.8.
Flush	None	Confirmation response?	Enables a caller to terminate all processing jobs and any that are pending; this is essentially a soft reset of the API. See Flush Method in Section 4.9.
GetMetrics	None	Metrics object	Enables a caller to perform a health check on the API, obtain any relevant statistics and so on, including how much load the API is currently experiencing. See GetMetrics Method in Section 4.10.

Method	Input	Output	Description
FetchMetaData	Token ID + (optional) row, count	Metadata collection	Enables the client to retrieve any predinfo metadata from the API. Also supports pagination, using an optional starting row and count parameters. See FetchMetaData Method in Section 4.11.

4.1 PGA Web Service Contract Diagram

This diagram describes the PGA web service contract:



High-level summary of the PGA web service contract

4.2 Contract Implementation

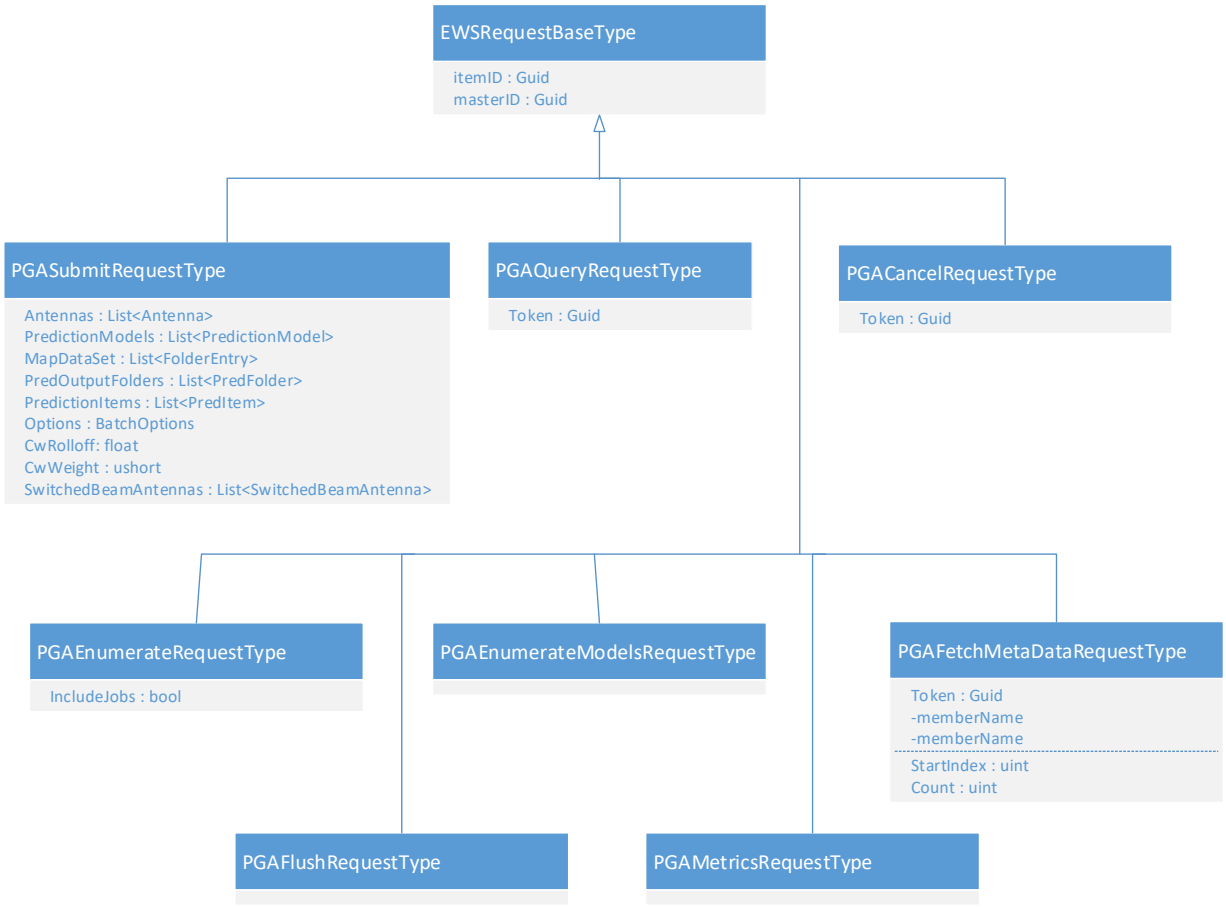
A simplified view of the service interface is provided in this picture to illustrate the operations:

Note: All operations use XmlSerializationFormat.

```
namespace TEOCO.PGA.Interfaces
{
    [ServiceContract(Namespace = NamespaceDefinitions.ContractsNamespace, Name =
"PGA.Service")]
```

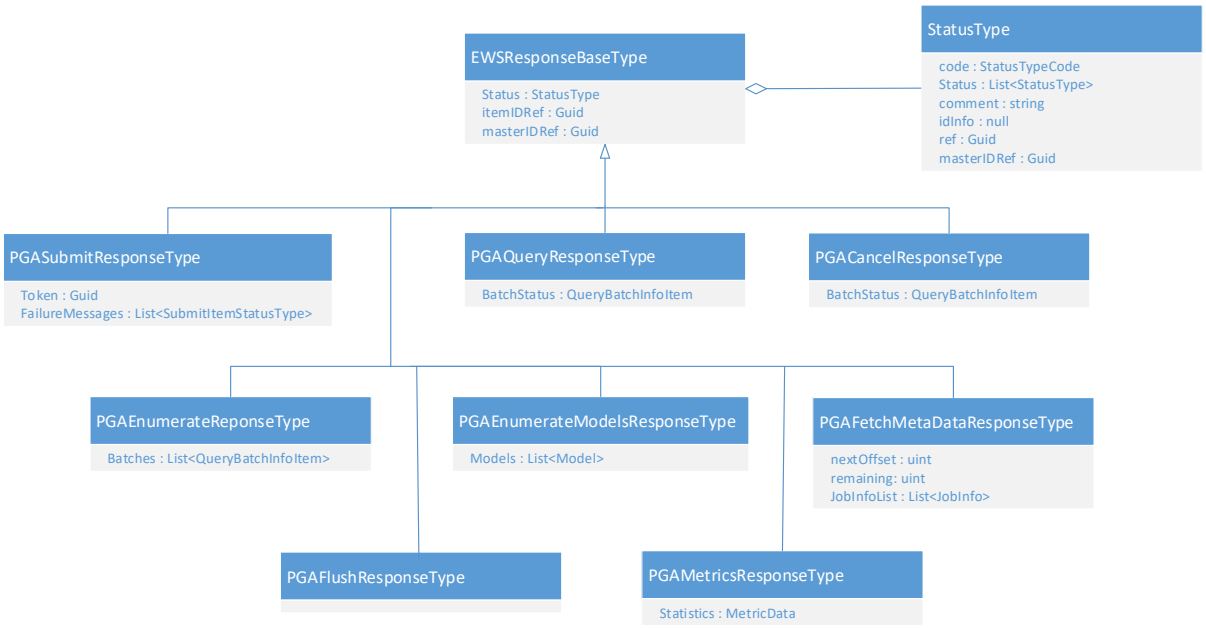
```
    [WsdlDocumentation("PGA Web Service 1.0")]
    [XmlSerializerFormat]
    public interface IPGAService
    {
        [OperationContract]
        PGASubmitResponseType Submit(PGASubmitRequestType request);

        [OperationContract]
        PGAQueryResponseType Query(PGAQueryRequestType request);

        [OperationContract]
        PGAEnumerateResponseType EnumerateBatches(PGAEnumerateRequestType request);

        [OperationContract]
        PGAEnumerateModelsResponseType EnumerateModels(PGAEnumerateModelsRequestType
request);

        [OperationContract]
        PGACancelResponseType Cancel(PGACancelRequestType request);

        [OperationContract]
        PGAFlushResponseType Flush(PGAFlushRequestType request);

        [OperationContract]
        PGAMetricsResponseType GetMetrics(PGAMetricsRequestType request);

        [OperationContract]
        PGAFetchMetadataResponseType FetchMetaData(PGAFetchMetadataRequestType
request);
    }
}
```

Each operation shares a common request and response base type, but also has elements which are specific to that operation. This picture shows both the common elements and the specific request types (at the top level only):



The response types also follow a similar pattern with some common elements (for example the same IDs specified on the original request), but all operations also set status information in addition to any operation-specific data returned.

This picture describes the responses (at the top level only):



4.3 Common Data Types

The details in this section are common to many of the methods. For that reason, they appear here in one place, instead of being repeated in several places.

- StatusTypeCode
- StatusType

4.3.1 StatusTypeCode

This is the status code for the requested operation. Possible values for each method call are listed here:

Method Call	Possible Values	Description or Comment
Submit	'OK'	All jobs sent successfully for processing.
	'Failed'	Batch validation or licence errors occurred.
	'Partial'	Some jobs were successfully sent for processing but some failed.
Query	'OK' or 'NoMoreObjects'	If the specified batch id in the 'Token' member of the request is unknown or has been cleared from the cache, then the Status.code member will return 'NoMoreObjects' rather than 'OK'.
EnumerateBatches	'OK' or 'NoMoreObjects'	If no batches are currently known to the service, then the service will return 'NoMoreObjects' rather than 'OK'.
EnumerateModels	'OK'	-
Cancel	'OK' or 'Failed'.	If the batch id specified in 'Token' is unknown or has been cleared from the service the Status.code member will be set to 'Failed'.
Flush	'OK'	-
GetMetrics	'OK'	-
FetchMetaData	'OK'	-

In addition, there are two more possible code values that are common to all method calls:

Method Call	Possible Values	Description or Comment
All	InvalidItemIDRef	For the appropriate request type (for example, PGASubmitRequestType, PGAQueryRequestType, etc.), the itemID was not specified, resulting in an empty Guid of 000000000-0000-0000-0000-0000000000.
All	UnexpectedError	Exception was thrown in the service and the state of the requested operation is unspecified.

4.3.2 StatusType

Parameter Name	Type	Description	Always Present
code	StatusTypeCode (Enumeration)	Status code for the requested operation. See Section 4.3.1.	Y
comment	String	Any relevant further information for the operation.	N
Status	StatusType (array)	List of status type objects.	N
idinfo	N/A	Not used.	N
ref	Guid	The identifier specified on the request.	Y
masterIDRef	Guid	The correlation identifier specified on the request, if it was included in the request.	Refer to Description column.

4.4 Submit Method

This method validates and processes the batch request information, breaks it down into jobs and passes all the data that is required to generate predictions. It immediately returns a 'token' that can be used for subsequent interactions with the PGA.

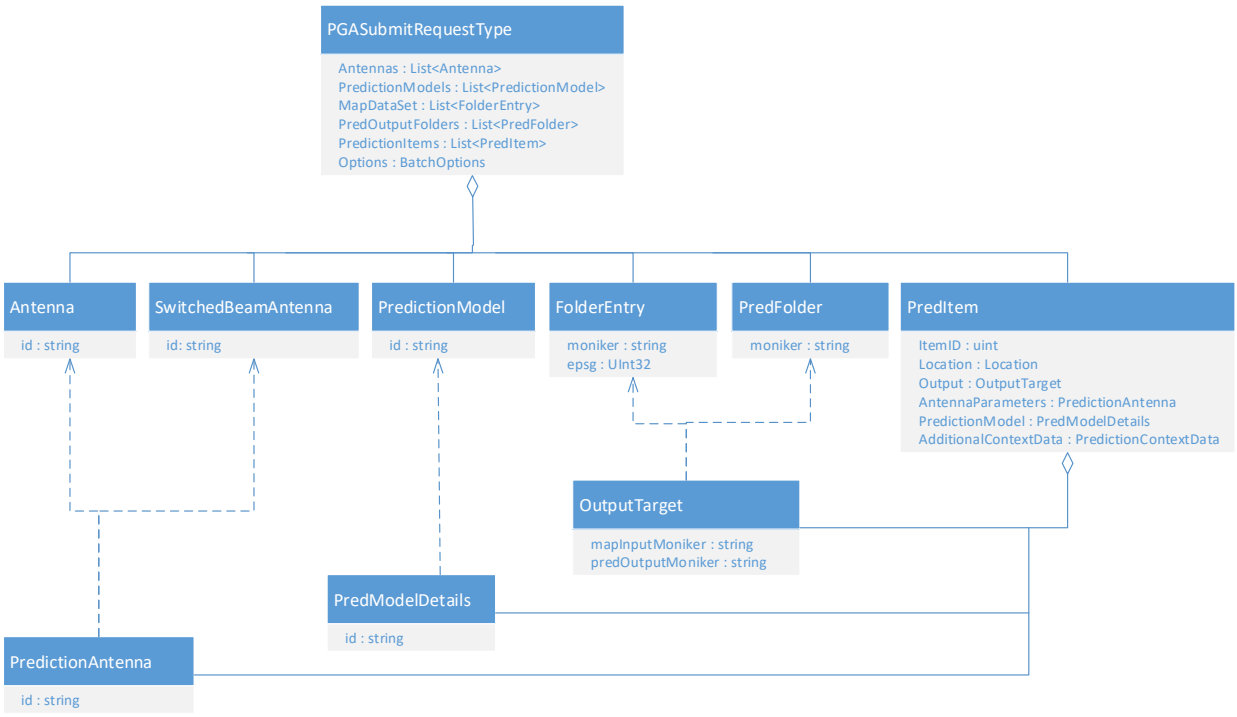
Notes:

- The service is asynchronous. On submission of the prediction request, the predictions are scheduled to be generated; you can use the return token to query when this has finished.
- Although the client receives a token, it is very likely that the effort required to create predictions will not have commenced immediately. The batch is queued until the resources to start processing the embedded job requests are available.

The **Submit** method call requires the following information:

Item	Equivalent Data Type Name
Cellular Antennas	**Antenna**
Switched Beam Antennas	**SwitchedBeamAntenna**
Propagation Models	**PredictionModel**
Map Data	**MapDataSet**
Folder to store the generated predictions	**OutputFolder**
Prediction Jobs	**PredictionJobItem**

This picture shows the structure of a Submit request (with only the key fields present in the classes; the full list is described in the following tables).



Notes:

- Some third-party propagation models require additional parameters in order to function, which are retrieved by the model when required using an 'info-grabber'. If any models referenced in the submit method use an info-grabber, then the individual jobs require extra context parameters; see 'AdditionalContextData' in the 'PredictionJobItem' table in Section 4.4.9.

  Depending on the implementation of the model, the absence of this information when required may either cause the prediction to fail outright. Even if it does not fail, it may be different to an equivalent prediction generated from within ASSET, which means that if the Prediction Access Module (PAM) is subsequently used, it will fail to locate the prediction correctly.

- The use of the 'PredictionContextData' type (see Section 4.4.8.5) is currently only relevant to the MYRIAD and Volcano models.

The following tables provide details of each of the data types.

4.4.1 Antenna

Parameter Name	Type	Description	Mandatory /Optional	Example Value
Id	String	The unique identifier for the antenna pattern. Example: 'Device1\Pattern1' **Note**: This is the parameter that is called by the id in the PredictionAntenna type. See PredictionAntenna in Section 4.4.8.3.	M	CMA_BDHH_6520_E0_8\CMA_BDHH_6520_E0_8_01D
Azimuth Offset	Double	The angle, expressed in degrees, between the Physical Azimuth bearing of an installed antenna and the maximum of its main beam in the azimuth plane. Azimuth Offset can be positive or negative.	M	20
TiltType	Enum: Electrical, Mechanical	Setting for Downtilt: Electrical or Mechanical	M	Electrical
Tilt	Double	Downtilt value (see above).	M	2.0
GainType	Enumeration	Represents the reference antenna. Allowable values: 'dBI' (isotropic) or 'dBD' (dipole).	M	dBI
Gain	Double	The gain for the antenna pattern.	M	1.0
Frequency	Double	The frequency band of the pattern.	M	25.0
FrontToBackRatio	Double	The front to back ratio of the pattern.	M	2.0
CrossPolarDiscrimination	Double	The loss caused by the use of cross-polar antennas.	M	1.5
PolarisationType	Enumeration	Allowable values: 'Horizontal' or 'Vertical' or 'CrossPolar' Represents the polarisation used for the pattern.	M	Horizontal
HorizontalMask	String	Pairs of values which combine to form the horizontal mask of the antenna pattern. For each pair, the value on the left represents the degrees (from 0° to 359.5°) and the value on the right represents the corresponding loss in dB. These losses are the reduction in antenna gain from the antenna pattern's point of maximum gain. For example: 0.00,0.0000;1.00,0.0000;2.00,0.1000 and so on.	M	{0,0;1,0;2,0.01;3,0.02}
VerticalMask	String	Pairs of values which combine to form the vertical mask of the antenna pattern. For each pair, the value on the left represents the degrees (from 0° to 359.5°) and the value on the right represents the corresponding loss in dB. These losses are the reduction in antenna gain from the antenna pattern's point of maximum gain. For example: 0.00,0.0000;1.00,0.0000;2.00,0.1000 and so on.	M	{0,0;1,0;2,0.01;3,0.02}

Parameter Name	Type	Description	Mandatory /Optional	Example Value
PatternBeamIndex	Int32	Unique value within the set of patterns within a specific Frequency.	Note: Only mandatory for Switched Beam Antennas. Otherwise it is ignored.	1

4.4.2 BeamIndex

Parameter Name	Type	Description	Mandatory /Optional	Example Value
index	Int32	This index is used to find matching 'PatternBeamIndex' values (see table above) to determine which patterns will be included in the beam set. See: SwitchedBeamAntenna, Section 4.4.4	M	1
beamType	Enumeration	Allowable values: 'Traffic', 'Control' or 'Both'.	M	Traffic
controlBeamIndex	Int32	If beamtype = C or B, then must be index 0-65534 Else is ignored (-1)	M	-1
trafficBeamIndex	Int32	If beamtype = T or B, then must be index 0-65534 Else is ignored (-1)	M	0

4.4.3 BeamSet

Parameter Name	Type	Description	Mandatory /Optional	Example Value
Id	String	The unique identifier for the beam set. **Note**: This is the parameter that is called by the SwitchedBeam_BeamSetMoniker in the PredictionAntenna type. See: PredictionAntenna, Section 4.4.8.3.	M	Set1
[Content]	BeamIndex[]	The set of beam index definitions	M	

4.4.4 SwitchedBeamAntenna

Parameter Name	Type	Description	Mandatory /Optional	Example Value
Id	String	The unique identifier for the antenna. Example: 'SBDevice1' **Note**: This is the parameter that is called by the id in the PredictionAntenna type. See: PredictionAntenna, Section 4.4.8.3.	M	AIR5331_B260_SB
BeamSets	BeamSet[]	The defined groups (sets) of patterns (beams) available on the antenna.	M	
Antennas	Antenna[]	The pattern definitions available on the antenna.	M	

4.4.5 PredictionModel

Parameter Name	Type	Description	Mandatory /Optional	Example Value
Id	String	Name of the propagation model. **Note**: This is the parameter that is called by the id in the PredModDetails type. See 'PredModelDetails' in Section 4.4.8.4.	M	Standard Long Range Model
ClassId	Guid	The unique identifier for the model, expressed as a randomly generated Guid.	See Note.	{520E6383-8177-4F47-9F7E-E9AF64E44ABE}
ProgId	String	The unique identifier for the model, determined by the model vendor.	See Note.	ManagedAircomPredModel.SLRModel
Params		Prediction Model Parameters **Note**: This data varies from model to model.	M	-

Note: ClassId and/or ProgId must be present. If both are present, the ClassId takes precedence.

4.4.6 MapDataSet

Parameter Name	Type	Description	Mandatory /Optional	Example Value
moniker	String	Name for this output folder (must be unique). **Note**: This is the parameter that is called by the mapInputMoniker in the OutputTarget type. See OutputTarget in Section 4.4.8.2.	M	Map01
Epsg	UInt32	The projected EPSG code for the coordinate system that will be used for generating the predictions. **Note**: If this is different from the EPSG specified for a prediction job, the PGA will perform the appropriate conversion from the job EPSG to this one. However if they are the same, the EPSG will be passed through unchanged.	M	32630
ClutterFolder	String	Path to folder containing the clutter data. **Important:** If you have enabled distribution on the PGA service, this path must be in UNC format.	M	\\Source\Clutter\Data
DTMFolder	String	Path to folder containing the height data. **Important:** If you have enabled distribution on the PGA service, this path must be in UNC format.	M	\\Source\Height\Data
BuildingRasterFolder	String	Path to folder containing the building raster data.	O	\\Source\BuildingRaster\Data

Parameter Name	Type	Description	Mandatory /Optional	Example Value
BuildingVectorFolder	String	Path to folder containing the building vector data.	O	\\Source\BuildingVector\Data

Note: More than one set of map data can exist – not all prediction jobs have to use the same set.

4.4.7 OutputFolder

Parameter Name	Type	Description	Mandatory /Optional	Example Value
moniker	String	Name for this output folder (must be unique). **Note**: This is the parameter that is called by the predOutputMoniker in the OutputTarget type. See OutputTarget in Section 4.4.8.2.	M	PredFolder1
Value	String	Path to folder which will containing the generated prediction files. **Important:** If you have enabled distribution on the PGA service, this path must be in UNC format.	M	\\dest\pred\output1

Note: More than one folder can exist – not all prediction jobs have to use the same output folder.

4.4.8 PredictionJobItem Components

These tables provide details of the constituent parts of the 'PredictionJobItem' data type. For details of the 'PredictionJobItem' data type itself, see Section 4.4.9.

4.4.8.1 Location

Parameter Name	Type	Description	Mandatory /Optional	Example Value
Epsg	UInt32	The EPSG for the coordinates specified for the antenna below. **Notes**: - If this differs from the EPSG specified from the map data level, then PGA will perform the appropriate conversion to the map data EPSG (if it is supported). However if they are the same, the EPSG will be passed through unchanged. - The service can accommodate a variety of EPSGs.	M	4269
X	Double	X coordinate for the antenna at the location for which you want to generate predictions. If the EPSG is that of a projected system (generally 5-digit EPSGs), then this value must be in cm. If the EPSG is that of a geodetic system (generally 4-digit EPSGs), then this value must be in degrees.	M	-117.8695
Y	Double	Y coordinate for the antenna at the location for which you want to generate predictions. If the EPSG is that of a projected system (generally 5-digit EPSGs), then this value must be in cm. If the EPSG is that of a geodetic system (generally 4-digit EPSGs), then this value must be in degrees.	M	48.914802
Z	Double	Z coordinate for the antenna, in cm. **Note:** This value is only required if you choose the override setting. For more information, see 'AllowLocationZOverride' in the 'Batch Options' table in Section 4.4.10.	See Note in previous column.	0

4.4.8.2 OutputTarget

Parameter Name	Type	Description	Mandatory /Optional	Example Value
mapInputMoniker	String	Name of map data folder which you want to be used for this prediction job. Note: ClassId and/or ProgId must be present. If both are present, the ClassId takes precedence. MapDataSet in Section 4.4.6.	M	Map01
predOutput Moniker	String	Name of output folder which you want to contain the prediction files for this job. See OutputFolder in Section 4.4.7.	M	PredFolder1

Note: The service can accommodate a variety of EPSGs for source and destinatio n coordinates, but there must be a valid coordinate transformation in order to be supported.

4.4.8.3 PredictionAntenna

Parameter Name	Type	Description	Mandatory /Optional	Example Value
Id	String	The unique identifier for the antenna pattern. See: Antenna, Section 4.4.1 - or - SwitchedBeamAntenna, Section 4.4.4	M	CMA_BDHH_6520_E0_8\CMA_BDHH_6520_E0_8_01D
Azimuth_Degrees	UInt32	The azimuth of the antenna (in degrees).	M	0
MechanicalTilt_Degrees	Double	Mechanical downtilt of the antenna.	M	0
Height_Cm	UInt64	Indicates the height of the antenna (in cm) that will be used for predictions.	M	3048
SwitchedBeam_Frequency	Double	The Frequency used to identify the subset of patterns from the Switched Beam antenna's complete set. This corresponds to the 'Frequency' field in the 'Antenna' data type. See: Antenna, Section 4.4.1	M Note: Only mandatory for Switched Beam antennas. Otherwise it is ignored.	1500
SwitchedBeam_BeamSet Moniker	String	The unique identifier for the beam set. See: SwitchedBeamAntenna, Section 4.4.4	As above.	Set1

4.4.8.4 PredModelDetails

Parameter Name	Type	Description	Mandatory /Optional	Example Value
Id	String	Name of the propagation model. See PredictionModel in Section 4.4.5.	M	UTM11_Spokane_2100_v4_11
Radius_Cm	Uint64	The radius for the predictions (in cm)	M	3500000
Resolution_Cm	Uint64	The resolution for the predictions (in cm)	M	2500

4.4.8.5 PredictionContextData

Important: The 'PredictionContextData' type contains fields required by the info-grabber associated with the target prediction model. The use of this type is currently only relevant to the MYRIAD and Volcano models.

Parameter Name	Type	Description	Mandatory /Optional	Example Value
AntennaDeviceName	String	The name of the antenna-device associated with the prediction antenna.	See note below.	Device1
AntennaPatternName	String	The name of the antenna-pattern associated with the prediction.	See note below.	Antenna1
FullyQualifiedAntennaPatternName	String	The fully qualified name of the antenna type of the prediction.	See note below.	Device1\Antenna1
AntennaPatternFrequency_Mhz	Single	The frequency of the antenna-pattern associated with the prediction, in MHz.	M	1062
AntennaPatternHash	Int64	A hash key unique to the antenna-pattern associated with the prediction.	M	001
LogicalAntennaPropertyName	String	The name of the property associated with the prediction antenna.	See note below.	Property01
LogicalAntennaIndex	Int32	The index of the logical antenna associated with the prediction.	See note below.	1
LogicalAntennaInstanceName	String	The (optional) instance-name of the logical antenna associated with the prediction.	See note below.	LogAntenna01
LogicalAntennaHash	Int64	A hash key unique to the logical-antenna associated with the prediction.	M	002
ModelHash	Int64	A hash key unique to the model associated with the prediction.	M	003
ModelInstanceName	String	The name of the model that will carry out the prediction if one is necessary. This is the name that would appear in ASSET's Propagation Models dialog.	See note below.	Standard Macrocell Model
NodeName	String	The name of the node associated with the prediction.	See note below.	Node01
NodeHash	Int64	A hash key unique to the source node associated with the prediction.	M	004
CellNames	String[]	A formatted list of names of the cells on the site of the prediction. Note that for predictions originating from sites using the GSM technology, there will only be a single name, but for other technologies there may be more than one.	See note below.	Cell01

Parameter Name	Type	Description	Mandatory /Optional	Example Value
IsPrimaryPrediction	Boolean	Indicates whether or not the targeted prediction represents the primary (as opposed to the secondary) prediction from the source node.	M	True

Note: The fields in the above table (such as AntennaDeviceName) marked with “See note below” are informational; they are passed to any info-grabber associated with the target prediction model. In principle they can be considered optional. However, the benefit is that they are used by the info-grabber to identify the source antenna-device in log messages, so it advisable to specify them.

4.4.9 PredictionJobItem

Parameter Name	Type	Description	Mandatory /Optional	Example Value
ItemID	UInt32	Unique identifier within the batch for the Prediction Job.	M	1
Location	Location	The source coordinate for the prediction job. See Section 4.4.8.1.	M	-
Output	OutputFolder	Describes the map data to use and the output folder for the prediction. See Section 4.4.7.	M	-
AntennaParameters	PredictionAntenna	The specified antenna equipment and required radius & resolution for the prediction. See Section 4.4.8.3.	M	-
PredictionModel	PredModelDetails	The specified model and required radius & resolution for the prediction. See Section 4.4.2.	M	-
AdditionalContextData	PredictionContextData	Additional ENTERPRISE specific information required only by some third party models to generate a correct ENTERPRISE based prediction. See Section 4.4.8.5.	O	-

4.4.10 BatchOptions

Available Options	Type	Brief Description	Example Value
None	String	No additional options will be used during predictions.	None
AllowHeightSmoothing	String	The prediction system can perform bilinear smoothing on the height data when predictions are created. This is only applicable for models that support smoothing of height data.	AllowHeightSmoothing
AllowPlcCorrection	String	The prediction system enables you to use measurement data to generate measurement-based pathloss correction files. **Note:** If you want to use this option, you must use the Prediction Access Module (PAM) or TEOCO's ENTERPRISE suite to populate the prediction output folder with the relevant correction file prior to generating the prediction.	AllowPlcCorrection
AllowCorrectionInterpolation	String	If you have selected the above measurement data option, you can also choose to use interpolation to influence surrounding pixels that do not contain measurement-based data. **Note:** You cannot use this option on its own *without* the AllowPlcCorrection option.	AllowCorrectionInterpolation
AllowLocationZOverride	String	If this override is not set, the Z coordinate (Height) is calculated based on the Height map data. If this override is set, you must provide this in the submitted data. See 'Location' table in Section 4.4.8.1.	AllowLocationZOverride

Note: You can activate multiple options by using space-separated strings.

4.4.11 PGASubmit

Parameter Name	Type	Description	Mandatory /Optional	Example Value
itemID	Guid	Unique identifier for the request specified by the client.	M	gh3bc006-98gb-55e3-018e-1e30ee21d126
masterID	Guid	Correlation identifier for a series of requests.	O	de1za884-78ez-33c3-896e-2e28cc91c904
Antennas	Antenna (array)	List of cellular antennas. See Section 4.4.1	O See Note 2 below.	-
PredictionModels	PredictionModel (array)	List of prediction models. See Section 4.4.5.	M	-
MapDataSet	MapDataSet (array)	List of map data folders. See Section 4.4.6.	M	-
PredOutputFolders	OutputFolder (array)	List of output folders. See Section 4.4.7.	M	-
PredictionItems	PredictionJobItem (array)	List of prediction jobs. See Section 4.4.8.	M	-
Options	BatchOptions	Selection of options to apply to all jobs. See Section 4.4.10	M	-
CwWeight	Ushort	Weight % for combining pathloss measurements with pathloss predictions.	O See Note 1 below.	-
CwRolloff	Float	Interpolation Rolloff factor for surrounding pixels.	O See Note 1 below.	-
SwitchedBeamAntennas	SwitchedBeamAntenna (array)	List of switched beam antennas. See Section 4.4.2	O See Note 2 below.	-

Note 1: CwWeight and CwRolloff should be included for Batch options that allow PLC corrections. These parameters depend on the AllowPlcCorrection and AllowCorrectionInterpolation options respectively. Please also see the information in the ‘BatchOptions’ table in Section 4.4.10.

Note 2: With regard to Antennas and SwitchedBeamAntennas, there must be a minimum of one ‘Passive’ cellular antenna, or one ‘Switched Beam’ cellular antenna, but there can also be both present.

The sections below correspond to the **Response Details** for the **Submit** method call:

4.4.12 StatusType

See ‘StatusType’ (within the Common Data Types section) in Section 4.3.2.

4.4.13 PGASubmitResponseType

Parameter Name	Type	Description	Always Present	Example Value
ItemID	Guid	The identifier specified on the request.	Y	gh3bc006-98gb-55e3-018e-1e30ee21d126
masterID	Guid	Correlation identifier for a series of requests. Note: This will only be present f it was included in the submitted request.	See Note in Description column.	de1za884-78ez-33c3-896e-2e28cc91c904
Status	StatusType	Overall status for the requested operation. See Section 4.3.2.	Y	Completed
Token	Guid	The service-generated unique identifier for the submitted prediction batch request.	Y (if successful)	ef2ab995-89fa-44d2-907d-0d29dd10c015

4.5 Query Method

This method allows you to query the summary status of a batch and its jobs. Summary information includes:

Batch:

- Status
- Submitted Time
- Planned Job Count
- Completed Job Count
- Completion Time

Jobs:

- ID
- Status
- Completion Time
- Message

4.5.1 PGAQueryRequestType

Parameter Name	Type	Description	Mandatory /Optional	Example Value
Token	Guid	The service-generated unique identifier for a previously submitted prediction batch request.	M	ef2ab995-89fa-44d2-907d-0d29dd10c015
itemID	Guid	Unique identifier for the request specified by the client.	M	gh3bc006-98gb-55e3-018e-1e30ee21d126
masterID	Guid	Correlation identifier for a series of requests.	O	de1za884-78ez-33c3-896e-2e28cc91c904

The sections below correspond to the **Response Details** for the **Query** method call:

4.5.2 QueryBatchJobInfoItem

Parameter Name	Type	Description	Always Present	Example Value
Id	UInt32	Unique identifier within the batch for the Prediction Job as specified on the submit call.	Y	ed7b336a-fd32-4389-b9dc-4059ce694b5f
Completed_Utc	DateTime	The time at which the job was finished. Note: Value only valid when job is finished.	See Note.	2017-07-05T09:31:32.5959151Z
Output Message	String	Feedback from the prediction engine. Note: Value only valid when job is finished.	See Note.	PredEngine: The prediction is available and up to date
Status	JobStatus (Enumeration)	Possible values: **Pending** - The job has not begun work yet and is idle. **Processing** - Work on the prediction for this job has begun. **Completed** - Work on the prediction for this job is complete and prediction is available. **Failed** - Work on the prediction failed and no prediction was generated. **Cancelled** - Work on the prediction was aborted and no prediction is available.	Y	Completed

4.5.3 QueryBatchInfoItem

Parameter Name	Type	Description	Always Present	Example Value
Id	Guid	The service-generated unique identifier for a previously submitted prediction batch request.	Y	ef2ab995-89fa-44d2-907d-0d29dd10c015
Status	BatchStatus (Enumeration)	Possible values: **Waiting** – no processing of any jobs in this batch has begun yet. **InProgress** – at least one job within the batch has begun. **Completed** – all jobs have completed processing (may not mean all succeeded). **Cancelling** – a cancel request for this batch has been received but has not yet completed – immediate abort of any remaining pending jobs and awaiting complete or abort of in progress jobs. **Cancelled** – this batch was cancelled but some or possibly even all jobs could have completed or aborted before the cancellation took effect.	Y	InProgress
Scheduled JobCount	Int64	Number of prediction jobs that were associated with this batch.	Y	10
CompletedJobCount	Int64	Number of prediction jobs that have finished (completed, cancelled or failed) for this batch. Note: Value changes as jobs finish.	Y	7
Submitted_Utc	DateTime	The time at which the batch was submitted.	Y	2017-07-05T09:31:29.1855741Z
Completed_Utc	DateTime	The time at which the entire batch was finished. Note: Value only valid when batch is finished.	See Note.	2017-07-05T09:32:5959151Z
Jobs	QueryBatchJobInfoItem (array)	Progress list for each prediction job in this batch. See Section 4.5.2.	N	-

4.5.4 StatusType

See ‘StatusType’ (within the Common Data Types section) in Section 4.3.2.

.

4.5.5 PGAQueryResponseType

Parameter Name	Type	Description	Always Present	Example Value
BatchInfo	QueryBatchInfoItem (array)	Progress report for the batch. See Section 4.5.3. Note: Value only valid upon success.	See Note.	-
Status	StatusType (array)	Overall status for the requested operation. See Section 4.3.2.	Y	-
itemIDRef	Guid	The identifier specified on the request.	Y	gh3bc006-98gb-55e3-018e-1e30ee21d126
masterIDref	Guid	The correlation identifier specified on the request, if it was included in the request.	N	de1za884-78ez-33c3-896e-2e28cc91c904

4.6 EnumerateBatches Method

This method provides a summary status per batch known to the service at the time the call is issued. Summary information includes:

- Status
- Submitted Time
- Planned Job Count
- Completed Job Count
- Completion Time (if finished)

4.6.1 PGAEnumerateRequestType

Parameter Name	Type	Description	Mandatory /Optional	Example Value
IncludeJobs	Boolean	'True' or 'False'	M	True
itemID	Guid	Unique identifier for the request specified by the client.	M	gh3bc006-98gb-55e3-018e-1e30ee21d126
masterID	Guid	Correlation identifier for a series of requests.	O	de1za884-78ez-33c3-896e-2e28cc91c904

The sections below correspond to the **Response Details** for the **EnumerateBatches** method call:

4.6.2 QueryBatchInfoItem

See the table in the 'Query Response Details' section.

4.6.3 QueryBatchJobInfoItem

See the table in the 'Query Response Details' section.

4.6.4 StatusType

See 'StatusType' (within the Common Data Types section) in Section 4.3.2.

4.6.5 PGAEnumerateResponseType

Parameter Name	Type	Description	Always Present	Example Value
Batches	QueryBatchInfoItem (array)	Progress reports for all known batches. See Section 4.5.3.	Y	-
Status	StatusType (array)	Overall status for the requested operation. See Section 4.3.2.	Y	-
itemIDRef	Guid	The identifier specified on the request.	Y	gh3bc006-98gb-55e3-018e-1e30ee21d126
masterIDref	Guid	The correlation identifier specified on the request, if it was included in the request.	N	de1za884-78ez-33c3-896e-2e28cc91c904

4.7 EnumerateModels Method

This method provides a list of the propagation models that are available to be used by the service. Summary information includes:

Note: This list will be applicable to Local Predictions only. It is not applicable to Distributed Predictions.

4.7.1 PGAEnumerateModelsRequestType

Parameter Name	Type	Description	Mandatory /Optional	Example Value
itemID	Guid	Unique identifier for the request specified by the client.	M	gh3bc006-98gb-55e3-018e-1e30ee21d126
masterID	Guid	Correlation identifier for a series of requests.	O	de1za884-78ez-33c3-896e-2e28cc91c904

The sections below correspond to the **Response Details** for the **EnumerateModels** method call:

4.7.2 PredictionModelDescription

Parameter Name	Type	Description	Always Present	Example Value
ClassId	Guid	The registered CLSID of the propagation model.	Y	35cdce55-c6c1-460b-8362-91ee9817ab8e
ProgId	String	The English descriptive registered COM ProgId of the propagation model.	N	Macrocell2.TUtilObject.10.0
Description	String	A vendor specified description of the propgation model.	Y	Standard Macrocell 2
HasInfoGrabber	Boolean	‘True’ or ‘False’ – indicates the model requires the use of an info-grabber and implies the requirement to specify the AdditionalContextData data for submit job requests.	Y	True

4.7.3 StatusType

See ‘StatusType’ (within the Common Data Types section) in Section 4.3.2.

.

4.7.4 PGAEnumerateModelsResponseType

Parameter Name	Type	Description	Always Present	Example Value
Models	PredictionModelDescription (array)	List of prediction models installed on the local machine. See Section 4.7.2.	Y	-
Status	StatusType (array)	Overall status for the requested operation. See Section 4.3.2.	Y	-
itemIDRef	Guid	The identifier specified on the request.	Y	gh3bc006-98gb-55e3-018e-1e30ee21d126
masterIDref	Guid	The correlation identifier specified on the request, if it was included in the request.	N	de1za884-78ez-33c3-896e-2e28cc91c904

4.8 Cancel Method

This method performs a cancellation at batch level. It allows cancellations to be sent for any pending jobs on the given batch.

The batch cancellation works as follows:

- If no job in the batch has started, it is cancelled immediately
- If any job in the batch has started but not completed, the job moves to ‘Cancelling’ state, and will be cancelled once that job reaches a completion state

4.8.1 PGACancelRequestType

Parameter Name	Type	Description	Mandatory /Optional	Example Value
Token	Guid	The service-generated unique identifier for a previously submitted prediction batch request.	M	ef2ab995-89fa-44d2-907d-0d29dd10c015
itemID	Guid	Unique identifier for the request specified by the client.	M	gh3bc006-98gb-55e3-018e-1e30ee21d126
masterID	Guid	Correlation identifier for a series of requests.	O	de1za884-78ez-33c3-896e-2e28cc91c904

The sections below correspond to the **Response Details** for the **Cancel** method call:

4.8.2 QueryBatchInfoItem

See the table in the ‘Query Response Details’ ’ section.

4.8.3 QueryBatchJobInfoItem

See the table in the ‘Query Response Details’ section.

4.8.4 StatusType

See ‘StatusType’ (within the Common Data Types section) in Section 4.3.2.

4.8.5 PGACancelResponseType

Parameter Name	Type	Description	Always Present	Example Value
BatchInfo	QueryBatchInfoItem	Progress report for the batch at the time the cancellation request was accepted. See Section 4.6.2. Note: Value only valid upon success.	See Note.	-
Status	StatusType	Overall status for the requested operation. See Section 4.3.2.	Y	-
itemIDRef	Guid	The identifier specified on the request.	Y	gh3bc006-98gb-55e3-018e-1e30ee21d126
masterIDRef	Guid	The correlation identifier specified on the request, if it was included in the request.	N	de1za884-78ez-33c3-896e-2e28cc91c904

4.9 Flush Method

This method allows cancellation of all pending jobs across all batches and resets the instance statistics.

Note: This method is asynchronous, therefore pending job notifications may show in the statistics immediately after flushing (Completed and Cancelled items are removed, but not Pending items).

4.9.1 PGAFlushRequestType

Parameter Name	Type	Description	Mandatory/ Optional	Example Value
itemID	Guid	Unique identifier for the request specified by the client.	M	gh3bc006-98gb-55e3-018e-1e30ee21d126
masterID	Guid	Correlation identifier for a series of requests.	O	de1za884-78ez-33c3-896e-2e28cc91c904

The sections below correspond to the **Response Details** for the **Flush** method call:

4.9.2 StatusType

See ‘StatusType’ (within the Common Data Types section) in Section 4.3.2.

4.9.3 PGAFlushResponseType

Parameter Name	Type	Description	Always Present	Example Value
Status	StatusType	Overall status for the requested operation. See Section 4.3.2.	Y	-
itemIDRef	Guid	The identifier specified on the request.	Y	gh3bc006-98gb-55e3-018e-1e30ee21d126
masterIDRef	Guid	The correlation identifier specified on the request, if it was included in the request.	N	de1za884-78ez-33c3-896e-2e28cc91c904

4.10 GetMetrics Method

This enables you to check on the progress of the service engine.

This method allows the retrieval of performance counter information from the running PGA instances. Performance measurements include:

- Number of batches received
- Number of batches rejected
- Number of batches completed
- Number of jobs received
- Number of jobs failed
- Number of jobs completed
- Number of duplicate jobs
- Number of jobs cancelled
- Min Batch Processing Time
- Avg Batch Processing Time
- Max Batch Processing Time

4.10.1 PGAMetricsRequestType

Parameter Name	Type	Description	Mandatory/ Optional	Example Value
itemID	Guid	Unique identifier for the request specified by the client.	M	gh3bc006-98gb-55e3-018e-1e30ee21d126
masterID	Guid	Correlation identifier for a series of requests.	O	de1za884-78ez-33c3-896e-2e28cc91c904

The sections below correspond to the **Response Details** for the **GetMetrics** method call:

4.10.2 ServiceVersionInfoType

Parameter Name	Type	Description	Always Present	Example Value
serviceName	String	The name of the service being described, as defined in the configuration file.	Y	TEOCO.PGA.Service.PGAServiceImpl
MajorVersionNumber	Int32	The major version number of the service, as defined in the configuration file.	N	1
MinorVersionNumber	Int32	The minor version number of the service, as defined in the configuration file.	N	0
ServiceEndPoints	String[]	List of available endpoint URLs to talk to the service, resolved according to the address configuration block from the configuration file. See Section 3.1.	Y	-

4.10.3 MetricData

Parameter Name	Type	Description	Always Present	Example Value
Statistics	StatisticsData	PGA-specific metric counters. See Section 4.10.4.	Y	-
ResponseTime	DateTime	The time at which this response was generated on the service (in server local time).	Y	2017-07-05T10:48:52.0773782 +01:00
MachineName	String	The hostname (if available) of the machine running the service.	Y	MSDN00552DT
ProcessName	String	The service's windows process name.	Y	TEOCO.PGA.Runtime
ProcessID	Int32	The PID of the windows process on the service hosting machine.	Y	3852
NonpagedSystemMemorySize	Int64	The amount of system memory, in bytes, allocated for the process that cannot be paged out to disk.	Y	110384
PagedMemorySize	Int64	The amount of memory, in bytes, allocated in the paged file on disk for this process.	Y	288198656
PagedSystemMemorySize	Int64	The amount of system memory, in bytes, allocated for the process that can be paged out to disk.	Y	514968

Parameter Name	Type	Description	Always Present	Example Value
PeakPagedMemorySize	Int64	The maximum amount of memory allocated by the process that could be paged out to disk.	Y	289972224
PeakVirtualMemorySize	Int64	The maximum amount of virtual memory, in bytes, allocated to the process since it was started.	Y	1050394624
PeakWorkingSet	Int64	The maximum amount of physical memory, in bytes, allocated to the process since it was started.	Y	104595456
StartTime	DateTime	The time at which the service was started on the service machine (in host local time).	Y	2017-07-05T09:46:56.0557325 +01:00
VirtualMemorySize	Int64	The amount of virtual memory, in bytes, currently allocated to the process.	Y	1041547264
WorkingSet	Int64	The amount of physical memory, in bytes, currently allocated to the process.	Y	101363712
AssemblyVersionInfo	String	The name of the executing assembly containing the definition of this data type.	Y	TEOCO.EWS.Framework, Version=4.0.0.0, Culture=neutral, PublicKeyToken=0a5f4b3188f9bbb0
ServiceVersioninfo	ServiceVersionInfoType	Provides information related to the web service(s) available and the list of endpoint URLs for communicating with it. See Section 4.10.2.	Y	-
InstanceName	String	Allows identification of originating instance for these statistics when multiple service instances are running on a single machine.	Y	Instance 1 [3852]
HostID	Guid	Reserved for future use.	Y	-

4.10.4 StatisticsData

Tip: The StatisticsData results can also be accessed as counters in the Windows Performance Monitor perfmon.

Parameter Name	Type	Description	Always Present	Example Value
NumberOfBatchRequestedReceived	Int64	Total number of batch requests received.	Y	15
NumberOfBatchRequessRefused	Int64	Total number of batch requests refused as being invalid (i.e. data errors).	Y	2
NumberOfBatchRequessCompleted	Int64	Total number of batch requests successfully completed work (includes cancellations).	Y	13
NumberOfJobsReceived	Int64	Total number of individual prediction jobs received across all batches.	Y	250
NumberOfJobsFailed	Int64	Total number of individual prediction jobs that failed to generate a prediction.	Y	7
NumberOfJobsSuccessful	Int64	Total number of individual prediction jobs that successfully completed (either a prediction was generated or it may have been the same as an existing prediction).	Y	243
NumberOfJobsDuplicates	Int64	Total number of individual prediction jobs that were actually duplicates of other prediction jobs. When PGA finds duplicate jobs, they are submitted in the normal way but are registered as duplicates, and attached to the 'original' job that is being processed. When the original job completes, the duplicate jobs are notified, and are all given the same status.	Y	10
NumberOfJobsCancelled	Int64	Total number of individual prediction jobs that were cancelled.	Y	0

4.10.5 StatusType

See 'StatusType' (within the Common Data Types section) in Section 4.3.2.

4.10.6 PGAMetricsResponseType

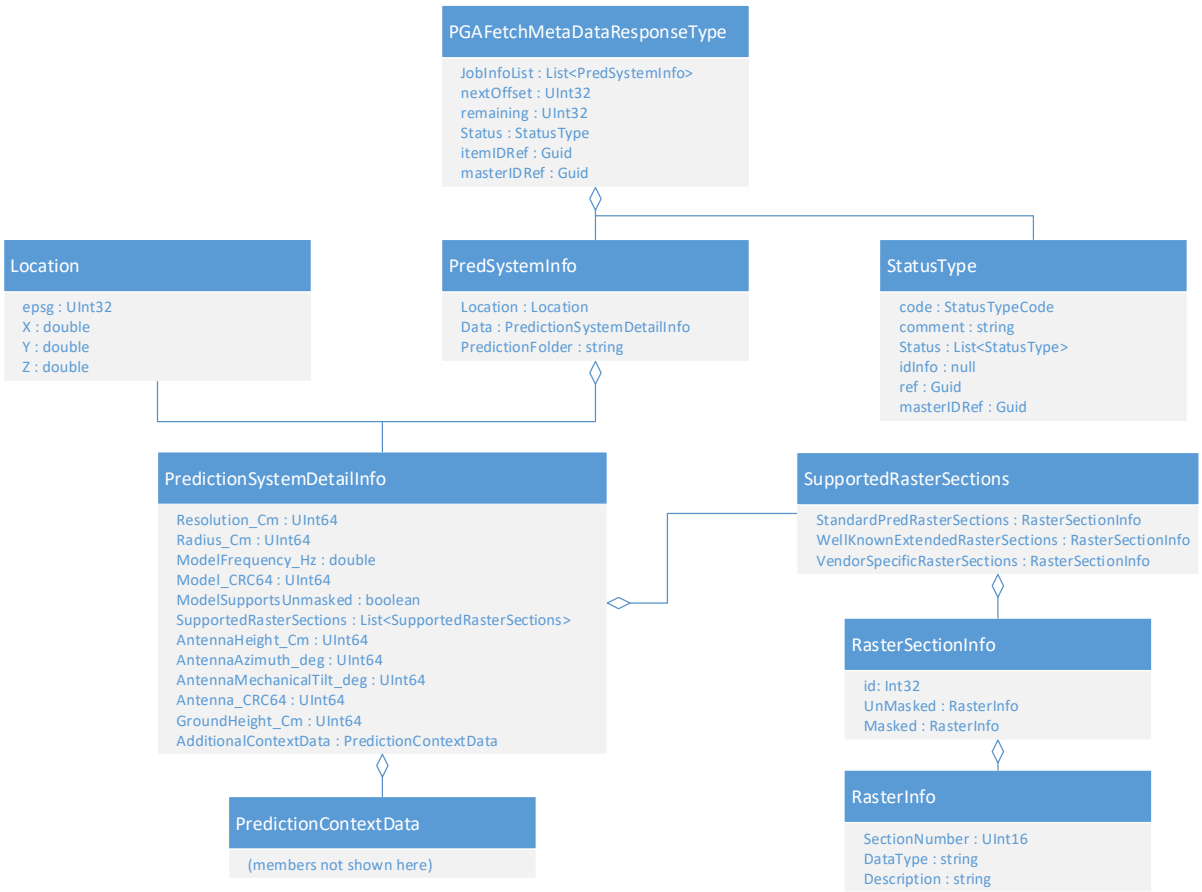
Parameter Name	Type	Description	Always Present	Example Value
Statistics	MetricData	Performance metrics for the current state of the service. See Section 4.10.3.	Y	-
Status	StatusType	Overall status for the requested operation. See Section 4.3.2.	Y	InProgress
itemIDRef	Guid	The identifier specified on the request.	Y	gh3bc006-98gb-55e3-018e-1e30ee21d126
masterIDRef	Guid	The correlation identifier specified on the request, if it was included in the request.	N	de1za884-78ez-33c3-896e-2e28cc91c904

4.11 FetchMetaData Method

This method allows the PGA service to generate sufficient information to be used to drive the Prediction Access Module (PAM).

Note: For more information, see the *Prediction Access Module Technical Reference Guide*.

This picture shows the structure of the response for a metadata request (with only the key fields present in the classes; the full list is described in the following tables).



The metadata is essentially the result of requesting a batch via the API.

Note: It is possible to retrieve metadata for completed jobs before the overall batch has completed.

4.11.1 PGAFetchMetadataRequestType

Parameter Name	Type	Description	Mandatory/ Optional	Example Value
Token	Guid	The service-generated unique identifier for a previously submitted prediction batch request.	M	ef2ab995-89fa-44d2-907d-0d29dd10c015
StartIndex	UInt32	The starting job index within a batch to paginate from.	M	3
Count	UInt32	The number of jobs within a page to return in the response. Note: If this is set to '0', the Count will default to the total number of jobs within the requested batch. But if it is set higher, it will never exceed that total number.	O	4
itemID	Guid	Unique identifier for the request specified by the client.	M	gh3bc006-98gb-55e3-018e-1e30ee21d126
masterID	Guid	Correlation identifier for a series of requests.	O	de1za884-78ez-33c3-896e-2e28cc91c904

The sections below correspond to the **Response Details** for the **FetchMetaData** method call:

4.11.2 RasterInfo

Parameter Name	Type	Description	Always Present	Example Value
SectionNumber	UInt16	The number of the raster section contained in stored prediction data featuring the described raster.	Y	2
DataType	String	The data-type (int/float; etc.) of the raster section in string form.	Y	float
Description	String	A text description of what the raster section represents (loss data; angle data; line of sight data; etc.)	Y	Angle data

4.11.3 RasterSectionInfo

Parameter Name	Type	Description	Always Present	Example Value
id	Int32	Section identifier for the prediction system.	Y	-
UnMasked	RasterInfo	Raster-info describing the un-masked raster form of the associated prediction data. See Section 4.11.2. Note: This is not generated by all prediction models, so may not be present.	N	-
Masked	RasterInfo	Raster-info describing the masked raster form of the associated prediction data. See Section 4.11.2.	Y	-

4.11.4 SupportedRasterSections

Parameter Name	Type	Description	Always Present	Example Value
StandardPredRasterSections	RasterSectionInfo list	A list of raster-section-info items describing all standard raster sections (e.g. pathloss) contained in the prediction data associated with the items from the associated batch. See Section 4.11.3.	Y	-
WellKnownExtendedRasterSections	RasterSectionInfo list	A list of raster-section-info items describing all well-known raster sections supported by the model (these are directly recognised by Asset and may be used if available, e.g. line-of-sight data) contained in the prediction data associated with the items from the associated batch. See Section 4.11.3.	N	-
VendorSpecificRasterSections	RasterSectionInfo list	A list of raster-section-info items describing all vendor-specific raster sections (these are defined by the third party model vendor and not directly recognised by Asset) contained in the prediction data associated with the items from the associated batch. See Section 4.11.3.	N	-

4.11.5 Location

Parameter Name	Type	Description	Always Present	Example Value
epsg	UInt32	The EPSG for the above coordinates. **Note**: The service can accommodate a variety of EPSGs.	Y	4277
X	Double	X coordinate for the antenna. If the EPSG is that of a projected system (generally 5-digit EPSGs), then this value must be in cm. If the EPSG is that of a geodetic system (generally 4-digit EPSGs), then this value must be in degrees.	Y	-117.8747306225478 3
Y	Double	Y coordinate for the antenna. If the EPSG is that of a projected system (generally 5-digit EPSGs), then this value must be in cm. If the EPSG is that of a geodetic system (generally 4-digit EPSGs), then this value must be in degrees.	Y	48.911034 739234132
Z	Double	Z coordinate for the antenna, in cm.	Y	0

4.11.6 PredictionContextData

Please see the table in the ‘PredictionContextData’ in Section 4.4.8.5.

4.11.7 PredictionSystemDetailInfo

Parameter Name	Type	Description	Always Present	Example Value
Resolution_Cm	UInt64	The prediction resolution in cm	Y	5000
Radius_Cm	UInt64	The prediction radius in cm	Y	1000000
ModelFrequency_Hz	Double	The propagation model frequency in Hz	Y	1800
Model_CRC64	UInt64	The generated CRC for all the propagation model parameters used for the prediction	Y	10175057394752411471
ModelSupportsUnmasked	Boolean	'True' or 'False' – indicates whether the model is capable of generating unmasked predictions	Y	true
Supported RasterSections	Supported RasterSections(array)	See Section 4.11.4.	Y	-
AntennaHeight_Cm	Int64	The height of the antenna itself (in cm).	Y	3048
AntennaAzimuth_deg	UInt64	The azimuth of the antenna in degrees	Y	0
AntennaMechanicalTilt_deg	Double	The mechanical tilt of the antenna in degrees	Y	0
Antenna_CRC64	UInt64	The generated CRC for all the antenna parameters used for the prediction	Y	1017505739475241171
GroundHeight_Cm	Int64	Either the calculated or manually specified value for the actual physical height of the antenna (in cm).	Y	5230
AdditionalContextData	PredictionContextData	If this information was supplied in the original prediction request, it will be provided here. See Section 4.11.6.	N	-

4.11.8 PredSystemInfo

Parameter Name	Type	Description	Always Present	Example Value
Id	UInt32	Unique identifier within the batch for the Prediction Job.	Y	1
Location	Location	The requested location of the prediction in the targeted coordinate system as defined by the job request. See Section 4.11.5.	Y	-
Data	PredictionSystemDetailInfo	Detailed parameters describing the prediction job.	Y	-
PredictionFolder	String	The specified output folder the prediction can be located in.	Y	PredFolder1
CwWeight	Ushort	Weight % for combining pathloss measurements with pathloss predictions.	Y	
CwRolloff	Float	Interpolation Rolloff factor for surrounding pixels.	Y	
Options	BatchOptions	Selection of options to apply to all jobs. See Section 4.4.10.	Y	
Flags	Ulong	PAM prediction flags.	N	

4.11.9 PGAFetchMetadataResponseType

Parameter Name	Type	Description	Always Present	Example Value
JobInfoList	PredSystemInfo (array)	List of prediction meta data information for each requested job. See Section 4.11.8.	Y	-
nextOffset	UInt32	Next job index after this request.	Y	3
remaining	UInt32	How many jobs remaining.	Y	4
Status	StatusType	Overall status for the requested operation. See Section 4.3.2.	Y	-
itemIDRef	Guid	The correlation identifier specified on the request, if it was included in the request.	Y	gh3bc006-98gb-55e3-018e-1e30ee21d126
masterIDRef	Guid	The identifier specified on the request.	N	de1za884-78ez-33c3-896e-2e28cc91c904

5 PGA REST Interface Description

PGA also provides a REST interface based upon the OpenAPI standard that supports authentication via declared api-key held within the PGA configuration (see section 3.1). This interface provides the same functionality as provided by the SOAP based Web Service described in the previous section, but adds the convenience of adding the features of PGA to more Web based clients.

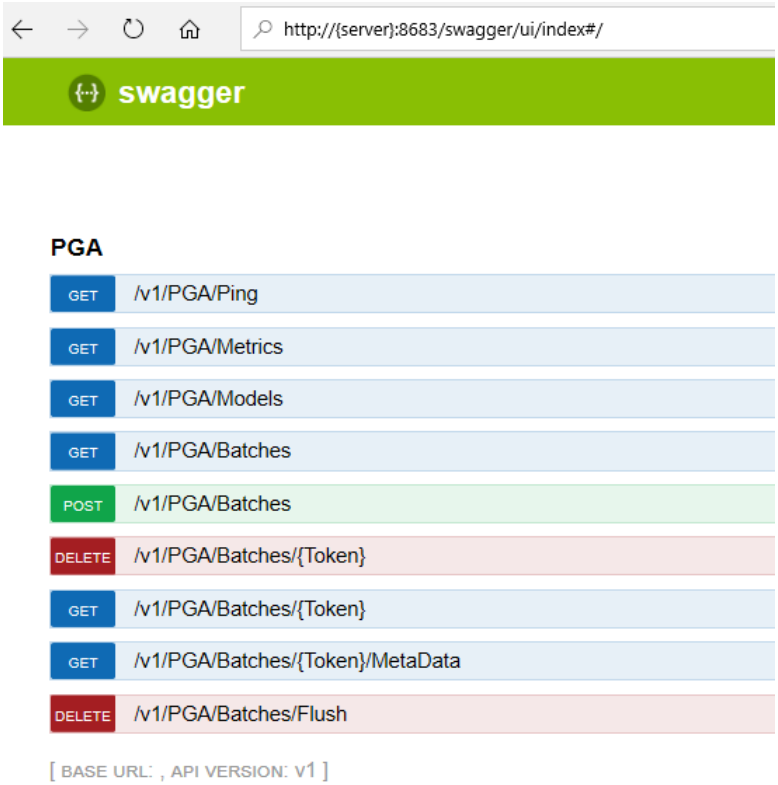
The data types used by the REST interface are identical to those used by the SOAP interface. Please refer to the relevant part of the 'PGA Web Service Description' chapter for the definitions of the various data types.

The new endpoint is available on the following port; this can be changed via the PGA configuration file:

`http://{server}:8683/`

5.1 Swagger UI

The OpenAPI standard includes a built-in mechanism to self-document its RESTful interfaces by default if you enter the above URL (where {server} is the name of your PGA server). You will be presented with the following web page:



From this user interface you can then navigate to the various supported RESTful methods that PGA offers and experiment or test with PGA functionality as you require. The interface will indicate the appropriate optional or mandatory parameters that the methods require and accept pasted input from the clipboard when requiring PGA data types.

Note: These methods are identical to those offered by the SOAP interface and will function similarly.

5.2 Working with JSON or XML data formats

In addition to providing the same functionality as the SOAP interface, the RESTful interface also allows the PGA data types to be provided in either JSON or XML format. The HTTP “Content-Type” header needs to be set to either “application/json” or “application/xml” for JSON or XML content. The following are examples of PGA data type provided in JSON and XML formats:

```
{
  "antennas": [
    {
      "id": "Default\\Default",
      "azimuthOffset": 0,
      "tiltType": "Electrical",
      "tilt": 0,
      "gainType": "dBI",
      "gain": 12.1,
      "frequency": 0,
      "frontToBackRatio": 0,
```

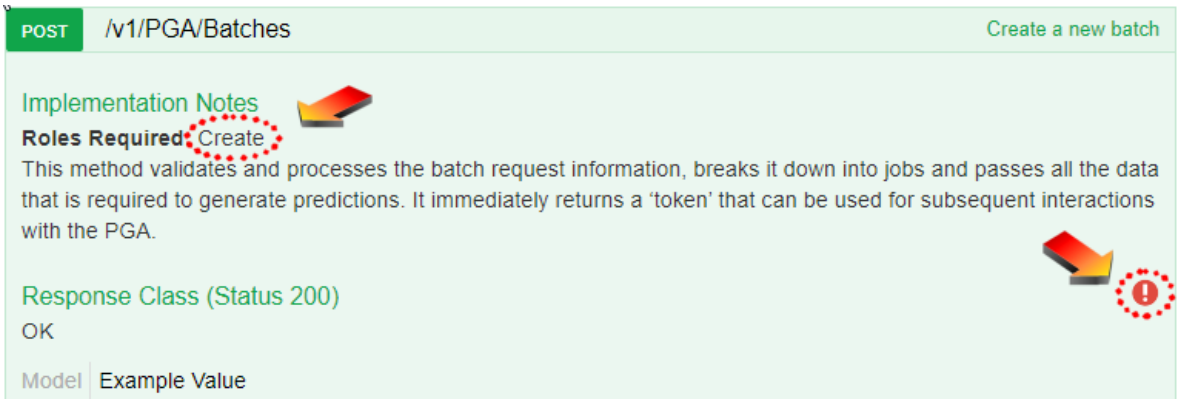
```
<?xml version="1.0"?>
<PGARestSubmitRequestType>
  <antennas>
    <id>Default\Default</id>
    <azimuthOffset>0</azimuthOffset>
    <tiltType>Electrical</tiltType>
    <tilt>0</tilt>
    <gainType>dBI</gainType>
    <gain>12.1</gain>
    <frequency>0</frequency>
    <frontToBackRatio>0</frontToBackRatio>
```

The format of the response can also be chosen by setting the HTTP header “Accept” to either “application/json” or “application/xml”. The response returned will be in the appropriate format.

5.3 Securing the RESTful Interface with API-key

By default, the RESTful interface will not permit any user who has not supplied an appropriate API key to the interface, to submit prediction jobs, or delete existing jobs. Only the Query methods can be used anonymously. See the Configuring Service Endpoints (section 3.1) for details on how to configure the api-keys.

From the Swagger UI, if you look more closely at the secured methods, you may note that each has a required role and a small red exclamation mark button that you can interact with:



If you select the red button, the UI will prompt you for an API-key. This value will then be cached by the Swagger UI until you refresh or close your web browser. Once the key has been established, the UI will change the colour of the exclamation to blue and allow you to use the “Try it out!!” button.

The API key is implemented as another HTTP header value named “api-key”, and can be supplied by the client applications where appropriate.

6 PGA Event Description

PGA can publish events via RabbitMQ to indicate the progress of batch jobs, and completion of prediction jobs. This allows for the clients to submit prediction jobs, and listen for the events that indicate the progress, and job completion status without needing to poll PGA at regular intervals for this information. This feature can be enabled and configured in the PGA configuration file (see section 3.3). Below are descriptions of the events published by the PGA, which can be consumed by the clients.

6.1 PGABatchEvent

This event indicates the progress of the batch of job. It is published whenever the batch status changes, or the number of completed jobs within the batch charges. Below is an example of this event:


```
{
      "data": {
             "server": "PGAServer",
             "batchId": "d0d642b2-b119-426e-ae50-9e1c40d0acd5",
             "status": "InProgress",
             "totalJobs": 11,
             "completedJobs": 7
      },
      "id": "8d47e89d-454f-4306-8c0f-a4c9289e3602",
      "ref": "a8282c61-ab53-4120-a175-dc9880b0d67a",
      "type": "PGABatchEvent",
      "created": "2019/10/18 11:46:08",
      "msgId": "teoco.events.asset",
      "msgHlp": "",
      "msgVer": "1.0"
}
```


Parameter Name	Type	Description
server	String	The hostname of the server that published the event.
batchId	Guid	The service-generated unique identifier for a previously submitted prediction batch request.
status	String	Current status of the batch. It can be one of Waiting, InProgress, Completed, Cancelled or Cancelling.
totalJobs	UInt32	Total number of jobs submitted in the batch.
completedJobs	UInt32	Number of jobs completed so far.
id	Guid	A unique message ID for the event
ref	Guid	A unique reference for the event
type	String	The event type
created	String	The timestamp of the event
msgId	String	A customisable string used to recognise PGA events
msgHlp	String	A customisable string used to indicate where online help might be located.
msgVer	String	A customisable string for versioning events

6.2 PGAJobCompleteEvent

This event is published as each of the prediction job completes, and indicates the success or failure of the prediction job. Below is an example of this event:


```
{
      "data": {
            "server": "PGAServer",
            "batchId": "d0d642b2-b119-426e-ae50-9e1c40d0acd5",
            "jobId": 2,
            "status": "Completed"
      },
      "id": "5b09b3b5-a019-475c-bc55-28bd829a59cf",
      "ref": "676ae44f-3c68-4869-a2bb-316365cbda22",
      "type": "PGAJobCompleteEvent",
      "created": "2019/10/18 11:46:08",
      "msgId": "teoco.events.asset",
      "msgHlp": "",
      "msgVer": "1.0"
}
```


Parameter Name	Type	Description
server	String	The hostname of the server that published the event.
batchId	Guid	The service-generated unique identifier for a previously submitted prediction batch request.
jobId	UInt32	Unique identifier within the batch for the Prediction Job.
status	String	Completion status of the prediction job. It can be one of Completed, Failed, or Cancelled.
id	Guid	A unique message ID for the event
ref	Guid	A unique reference for the event
type	String	The event type
created	String	The timestamp of the event
msgId	String	A customisable string used to recognise PGA events
msgHlp	String	A customisable string used to indicate where online help might be located.
msgVer	String	A customisable string for versioning events

7 Using the PGA Loader

The PGA Loader is a relatively simple command line interface which is capable of submitting batches (prediction submissions and associated query operations) to the PGA service, using XML data files. It is useful for simple integrations with the API or for diagnosis of possible system faults.

This chapter details the command line syntax for each operation that the Loader supports:

Command	Described in section	Associated Web Service Method
Submit	7.1	Submit
Query	7.2	Query
List	7.3	EnumerateBatches
Models	7.4	EnumerateModels
Cancel	7.5	Cancel
Flush	7.6	Flush
Stats	7.7	GetMetrics
Metadata	7.8	FetchMetadata

Note: In the syntax shown for each operation, [...] indicates an optional parameter.

7.1 Submit Command

The Submit command invokes the Submit method on the web service. See Submit Method in Section 4.4.

You can perform a batch submission using component parts, or a batch submission using a whole file.

To perform a batch submission using component parts, use the following syntax:

```
-submit  [-antennas=(file)]
         [-swbantennas=(file)]
         [-models=(file)]
         [-outputfolders=(file)]
         [-mapdatafolders=(file)]
         [-jobs=(file)]
         [-options="(option1),(option2),(option3)"]
         [-results=(file)]
         [-server=(hostname|IP address)]
         [-pollinterval=(seconds)]
         [-pollcount=(count)]
```

Where:

Parameter	Description
antennas	The name of an XML file containing a set of 'Passive' cellular antenna definitions to include in the batch. For an example, see Example Antennas File (Passive) in Section 8.1.
swbantennas	The name of an XML file containing a set of 'Switched Beam' cellular antenna definitions to include in the batch. For an example, see Example Antennas File (Switched Beam) in Section 8.2.
models	The name of an XML file containing a set of prediction model definitions to include in the batch. For an example, see Example Models File in Section 8.3.
outputfolders	The name of an XML file containing a set of output folder descriptions to include in the batch. For an example, see Example Output Folder File in Section 8.5.
mapdatafolders	The name of an XML file containing a set of map data folder descriptions to include in the batch. For an example, see Example Map Data Folder File in Section 8.4.
jobs	The name of an XML file containing a set of prediction job items that define the batch work. For an example, see Example Jobs File in Section 8.6.
options	A comma-separated string of one or more of the following options to apply to the batch: • AllowHeightSmoothing • AllowPlcCorrection • AllowCorrectionInterpolation • AllowLocationZOverride If you want to use more than one option, use commas to separate them (as in the example that follows this table). The AllowPlcCorrection and AllowCorrectionInterpolation options relate to the cwweight and cwrolloff parameters (see note at bottom of table). **Note**: For full information on these options, see the 'BatchOptions' table in Section 4.4.10.
results	By default, the PGA Loader returns the results directly into the command line. However, if provided, this parameter returns the results of the submission into the specified XML file.
server	If provided, this parameter directs the loader to use the specified hostname or IP address hosting a PGA instance rather than the default of 'localhost'. The other parts of the endpoint mapping must be the same on the other machine as configured in the app.config file for the loader.
pollinterval	Defines the polling interval in seconds between query status attempts on a submitted batch. If this is not specified, the batch is submitted immediately.
pollcount	Defines the number of polling attempts to make in total when performing query status attempts on a submitted batch. If this parameter is present, then 'pollinterval' must also be present.
cwweight	Weight % for combining pathloss measurements with pathloss predictions. Relates to the AllowPlcCorrection option.
cwrolloff	Interpolation Rolloff factor for surrounding pixels. Relates to the AllowCorrectionInterpolation option.

Note: CwWeight and CwRolloff should be included for Batch options that allow PLC corrections. These parameters depend on the AllowPlcCorrection and AllowCorrectionInterpolation options respectively. Please also see the information in the 'BatchOptions' table in Section 4.4.10.

This shows an example:

```
-submit  -antennas=PGA Data\antennas.xml
         -swbantennas=PGA Data\swb_antennas.xml
         -models=PGA Data\models.xml
         -outputfolders=PGA Data\outputfolders.xml
         -mapdatafolders=PGA Data\mapdataset.xml
         -jobs=PGA Data\jobs.xml
        [-options=AllowHeightSmoothing,AllowLocationZOverride]
        [-results=PGA Data\results.xml]
        [-server=PGAsvr01| 001.001.01.001]
        [-pollinterval=2]
        [-pollcount=3]
```

To perform a batch submission using a whole file, use the following syntax:

```
-submit  -batch=(file)
        [-results=(file)]
        [-server=(hostname|IP address)]
        [-pollinterval=(seconds)]
        [-pollcount=(count)]
```

Where:

Parameter	Description
batch	The name of an XML file containing the full content of a batch including antennas, models, folders, jobs, etc. For an example, see Example Batch File in Section 8.7.
results	By default, the PGA Loader returns the results directly into the command line. However, if provided, this parameter returns the results of the submission into the specified XML file.
server	If provided, this parameter directs the loader to use the specified hostname or IP address hosting a PGA instance rather than the default of 'localhost'. The other parts of the endpoint mapping must be the same on the other machine as configured in the app.config file for the loader.
pollinterval	Defines the polling interval in seconds between query status attempts on a submitted batch.
pollcount	Defines the number of polling attempts to make in total when performing query status attempts on a submitted batch. If this parameter is present, then 'pollinterval' must also be present.

This shows an example command:

```
-submit      -batch=PGA Data\batch.xml
             -results=PGA Data\results.xml
             -server= PGAsvr01| 001.001.01.001
             -pollinterval=2
             -pollcount=3
```

7.2 Query Command

The Query command invokes the Query method on the web service. See Query Method in Section 4.5.

To perform a query to retrieve the status of a submitted batch, use the following syntax:

```
-query  -token=(guid)
       [-results=(file)]
       [-server=(hostname|IP address)]
```

Where:

Parameter	Description
token	A Guid representing an identifier of the batch that you want to query.
results	By default, the PGA Loader returns the results directly into the command line. However, if provided, this parameter returns the results of the submission into the specified XML file.
server	If provided, this parameter directs the loader to use the specified hostname or IP address hosting a PGA instance rather than the default of 'localhost'. The other parts of the endpoint mapping must be the same on the other machine as configured in the app.config file for the loader.

This shows an example command:

```
-query  -token=2fce594c-12eb-46e0-a23d-538c781ba655
        -results=PGA Data\results.xml
        -server= PGAsvr01| 001.001.01.001
```

This shows an example of what is returned:

```
C:\Program Files\TEOCO\ENTERPRISE 10.0\Common>TEOCO.PGA.Loader -query -token=ef2a
b995-89fa-44d2-907d-0d29dd10c015

Product Version: 1.0.0.0
File Version:    1.0.0.0

<?xml version="1.0" encoding="ibm850"?>
<QueryBatchInfoItem xmlns:ews="http://www.aircominternational.com/contract/EWS/2
011/01" xmlns:pga="http://www.teoco.com/pga/contracts/2017/03" xmlns:xsi="http:/
/www.w3.org/2001/XMLSchema-instance" id="ef2ab995-89fa-44d2-907d-0d29dd10c015">
  <pga:BatchStatus>Completed</pga:BatchStatus>
  <pga:ScheduledJobCount>1</pga:ScheduledJobCount>
  <pga:CompletedJobCount>1</pga:CompletedJobCount>
  <pga:Submitted_Utc>2017-07-05T09:31:29.1855741Z</pga:Submitted_Utc>
  <pga:Completed_Utc>2017-07-05T09:31:32.5959151Z</pga:Completed_Utc>
  <pga:Jobs>
    <pga:Job id="0" status="Completed">
      <pga:Completed_Utc>2017-07-05T09:31:32.5929148Z</pga:Completed_Utc>
      <pga:OutputMessage>PredEngine: Success</pga:OutputMessage>
    </pga:Job>
  </pga:Jobs>
</QueryBatchInfoItem>
05/07/2017 10:42:20 : -query -token=ef2ab995-89fa-44d2-907d-0d29dd10c015
05/07/2017 10:42:20 : Elapsed time: 00:00:00.78
```

7.3 List Command

The List command invokes the EnumerateBatches method on the web service. See EnumerateBatches Method in Section 4.6.

To perform a query to retrieve the status of all current known batches, use the following syntax:

```
-list [-includejobs]
      [-results=(file)]
      [-server=(hostname|IP address)]
```

Where:

Parameter	Description
includejobs	Returns specific results for each job in the batch – id, status, time completed and the associated output message.
results	By default, the PGA Loader returns the results directly into the command line. However, if provided, this parameter returns the results of the submission into the specified XML file.
server	If provided, this parameter directs the loader to use the specified hostname or IP address hosting a PGA instance rather than the default of 'localhost'. The other parts of the endpoint mapping must be the same on the other machine as configured in the app.config file for the loader.

This shows an example command:

```
-list  -includejobs
       -results=PGA Data\results.xml
       -server= PGAsvr01| 001.001.01.001
```

This shows an example of what is returned:

```
C:\Program Files\TEOCO\ENTERPRISE 10.0\Common>TEOCO.PGA.Loader -list

Product Version: 1.0.0.0
File Version:    1.0.0.0

<?xml version="1.0" encoding="ibm850"?>
<ArrayOfQueryBatchInfoItem xmlns:ews="http://www.aircominternational.com/contrac
t/EWS/2011/01" xmlns:pga="http://www.teoco.com/pga/contracts/2017/03" xmlns:xsi=
"http://www.w3.org/2001/XMLSchema-instance">
  <QueryBatchInfoItem id="ef2ab995-89fa-44d2-907d-0d29dd10c015">
    <pga:BatchStatus>Completed</pga:BatchStatus>
    <pga:ScheduledJobCount>1</pga:ScheduledJobCount>
    <pga:CompletedJobCount>1</pga:CompletedJobCount>
    <pga:Submitted_Utc>2017-07-05T09:31:29.1855741Z</pga:Submitted_Utc>
    <pga:Completed_Utc>2017-07-05T09:31:32.5959151Z</pga:Completed_Utc>
    <pga:Jobs />
  </QueryBatchInfoItem>
</ArrayOfQueryBatchInfoItem>
05/07/2017 10:33:14 : -list
05/07/2017 10:33:14 : Elapsed time: 00:00:00.79
```

7.4 Models Command

The Models command invokes the EnumerateModels method on the web service. See EnumerateModels Method in Section 4.7.

To perform a query to retrieve all current known propagation models installed on this machine, use the following syntax:

```
-models [-results=(file)]
        [-server=(hostname|IP address)]
```

Where:

Parameter	Description
results	By default, the PGA Loader returns the results directly into the command line. However, if provided, this parameter returns the results of the submission into the specified XML file.
server	If provided, this parameter directs the loader to use the specified hostname or IP address hosting a PGA instance rather than the default of 'localhost'. The other parts of the endpoint mapping must be the same on the other machine as configured in the app.config file for the loader.

Note: This is only relevant to local predictions; if you are distributing the predictions, then the Distribution Coordinator determines the available models.

This shows an example command:

```
-models   -results=PGA Data\results.xml
          -server= PGAsvr01| 001.001.01.001
```

This shows an example of what is returned:

```
/contract/EWS/2011/01" xmlns:pga="http://www.teoco.com/pga/contracts/2017/03" xm
lns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <PredictionModelDescription>
    <pga:ClassId>35cdce55-c6c1-460b-8362-91ee9817ab8e</pga:ClassId>
    <pga:ProgId>Macrocell2.TUtilObject.9.1</pga:ProgId>
    <pga:Description>Standard Macrocell 2</pga:Description>
    <pga:HasInfoGrabber>false</pga:HasInfoGrabber>
  </PredictionModelDescription>
  <PredictionModelDescription>
    <pga:ClassId>3d37126a-8011-49ae-8c67-ab22b8915667</pga:ClassId>
    <pga:ProgId>FreeSpaceModel.CFreeSpaceModelModuleModule.9.1</pga:ProgId>
    <pga:Description>Free Space Loss Model</pga:Description>
    <pga:HasInfoGrabber>false</pga:HasInfoGrabber>
  </PredictionModelDescription>
  <PredictionModelDescription>
    <pga:ClassId>4d6e64a4-dcc3-4dbb-8115-0c3319737e4e</pga:ClassId>
    <pga:ProgId>ManagedAircomPredModel.SUIModel</pga:ProgId>
    <pga:Description>IEEE 802.16 SUI Model V1</pga:Description>
    <pga:HasInfoGrabber>false</pga:HasInfoGrabber>
  </PredictionModelDescription>
  <PredictionModelDescription>
    <pga:ClassId>59ac88b2-62eb-4b38-88e9-164e6ff93eef</pga:ClassId>
    <pga:ProgId>Macrocell.TUtilObject.9.1</pga:ProgId>
    <pga:Description>Standard Macrocell</pga:Description>
    <pga:HasInfoGrabber>false</pga:HasInfoGrabber>
  </PredictionModelDescription>
  <PredictionModelDescription>
    <pga:ClassId>67b14f69-b1b1-428d-8b39-e8810a047db4</pga:ClassId>
    <pga:ProgId>ManagedAircomPredModel.SLRModel</pga:ProgId>
    <pga:Description>Standard Long Range Model v1</pga:Description>
    <pga:HasInfoGrabber>false</pga:HasInfoGrabber>
  </PredictionModelDescription>
```

7.5 Cancel Command

The Cancel command invokes the Cancel method on the web service. See Cancel Method in Section 4.8.

To cancel any pending work on a submitted batch, use the following syntax:

```
-cancel  -token=(guid)
        [-results=(file)]
        [-server=(hostname|IP address)]
```

Where:

Parameter	Description
token	A Guid representing an identifier of the batch that you want to cancel.
results	By default, the PGA Loader returns the results directly into the command line. However, if provided, this parameter returns the results of the submission into the specified XML file.
server	If provided, this parameter directs the loader to use the specified hostname or IP address hosting a PGA instance rather than the default of 'localhost'. The other parts of the endpoint mapping must be the same on the other machine as configured in the app.config file for the loader.

This shows an example:

```
-cancel  -token=2fce594c-12eb-46e0-a23d-538c781ba655
         -results=PGA Data\results.xml
         -server= PGAsvr01| 001.001.01.001
```

7.6 Flush Command

The Flush command invokes the Flush method on the web service. See Flush Method in Section 4.9.

To instruct the service to cancel any pending batches and reset itself, use the following syntax:

```
-flush [-server=(hostname|IP address)]
```

Where:

Parameter	Description
server	If provided, this parameter directs the loader to use the specified hostname or IP address hosting a PGA instance rather than the default of 'localhost'. The other parts of the endpoint mapping must be the same on the other machine as configured in the app.config file for the loader.

This shows an example command:

```
-flush  -server= PGAsvr01| 001.001.01.001
```

This shows an example of what is returned:

```
C:\Program Files\TEOCO\ENTERPRISE 9.1\Common>TEOCO.PGA.Loader -flush

Product Version: 1.0.0.0
File Version:    1.0.0.0

Service accepted command to flush all jobs and reset

05/07/2017 10:51:53 : -flush
05/07/2017 10:51:53 : Elapsed time: 00:00:00.71
```

7.7 Stats Command

The Stats command invokes the GetMetrics method on the web service. See GetMetrics Method in Section 4.10.

To perform a query on the service to retrieve its metrics, use the following syntax:

```
-stats [-server=(hostname|IP address)]
```

Where:

Parameter	Description
server	If provided, this parameter directs the loader to use the specified hostname or IP address hosting a PGA instance rather than the default of 'localhost'. The other parts of the endpoint mapping must be the same on the other machine as configured in the app.config file for the loader.

This shows an example command:

```
-stats  -server= PGAsvr01| 001.001.01.001
```

This shows an example of what is returned:

```
<?xml version="1.0" encoding="ibm850"?>
<MetricData xmlns:ews="http://www.aircominternational.com/contract/EWS/2011/01"
xmlns:pga="http://www.teoco.com/pga/contracts/2017/03" xmlns:xsi="http://www.w3.
org/2001/XMLSchema-instance" HostID="00000000-0000-0000-0000-000000000000">
  <ResponseTime>2017-07-05T10:48:52.0773782+01:00</ResponseTime>
  <MachineName>MSDN00552DT</MachineName>
  <ProcessName>TEOCO.PGA.Runtime</ProcessName>
  <ProcessID>3852</ProcessID>
  <NonpagedSystemMemorySize>110384</NonpagedSystemMemorySize>
  <PagedMemorySize>288198656</PagedMemorySize>
  <PagedSystemMemorySize>514968</PagedSystemMemorySize>
  <PeakPagedMemorySize>289972224</PeakPagedMemorySize>
  <PeakVirtualMemorySize>1050394624</PeakVirtualMemorySize>
  <PeakWorkingSet>104595456</PeakWorkingSet>
  <StartTime>2017-07-05T09:46:56.0557325+01:00</StartTime>
  <VirtualMemorySize>1041547264</VirtualMemorySize>
  <WorkingSet>101363712</WorkingSet>
  <AssemblyVersionInfo>TEOCO.EWS.Framework, Version=4.0.0.0, Culture=neutral, Pu
blicKeyToken=0a5f4b3188f9bbb0</AssemblyVersionInfo>
  <ServiceVersionInfo />
  <InstanceName>Instance 1 [3852]</InstanceName>
  <Statistics>
    <NumberOfBatchRequestedRecieved>1</NumberOfBatchRequestedRecieved>
    <NumberOfBatchRequestsRefused>0</NumberOfBatchRequestsRefused>
    <NumberOfBatchRequestsCompleted>1</NumberOfBatchRequestsCompleted>
    <NumberOfJobsRecieved>1</NumberOfJobsRecieved>
    <NumberOfJobsFailed>0</NumberOfJobsFailed>
    <NumberOfJobsSuccessful>1</NumberOfJobsSuccessful>
    <NumberOfJobsDuplicates>0</NumberOfJobsDuplicates>
    <NumberOfJobsCancelled>0</NumberOfJobsCancelled>
  </Statistics>
</MetricData>
```

7.8 Metadata Command

The Metadata command invokes the FetchMetaData method on the web service. See FetchMetaData Method in Section 4.11.

To perform a query for a submitted batch to return the **metadata** required to drive the Prediction Access Module, use the following syntax:

```
-metadata  -token=(guid)
          [-index=(value)]
          [-count=(value)]
          [-results=(file)]
          [-server=(hostname|IP address)]
```

Where:

Parameter	Description
token	A Guid representing an identifier of the batch for which you want to retreive the metadata.
index	The starting job index within a batch to paginate.
count	The number of jobs within a page to return in the response.
results	By default, the PGA Loader returns the results directly into the command line. However, if provided, this parameter returns the results of the submission into the specified XML file. **Tip:** Due to the large amount of data that this command may generate, it is recommended to specify this parameter.
server	If provided, this parameter directs the loader to use the specified hostname or IP address hosting a PGA instance rather than the default of 'localhost'. The other parts of the endpoint mapping must be the same on the other machine as configured in the app.config file for the loader.

This shows an example command:

```
-metadata  -token=2fce594c-12eb-46e0-a23d-538c781ba655
           -index=4
           -count=5
           -results=PGA Data\results.xml
           -server= PGAsvr01| 001.001.01.001
```

This shows an example XML results file:

```
<?xml version="1.0" encoding="utf-8"?>
<ArrayOfPredSystemInfo xmlns:ews="http://www.aircominternational.com/contract/E
  <PredSystemInfo>
    <pga:Location epsg="4277">
      <pga:X>-117.87473062254783</pga:X>
      <pga:Y>48.911034739234132</pga:Y>
      <pga:Z>0</pga:Z>
    </pga:Location>
    <pga:Data>
      <pga:Resolution_Cm>5000</pga:Resolution_Cm>
      <pga:Radius_Cm>1000000</pga:Radius_Cm>
      <pga:ModelFrequency_Hz>1800</pga:ModelFrequency_Hz>
      <pga:Model_CRC64>10175057394752411471</pga:Model_CRC64>
      <pga:ModelSupportsUnmasked>true</pga:ModelSupportsUnmasked>
      <pga:SupportedRasterSections>
        <pga:StandardPredRasterSections>
          <pga:Section id="0">
            <pga:UnMasked>
              <pga:SectionNumber>0</pga:SectionNumber>
              <pga:DataType>RasterSectionType_Float</pga:DataType>
              <pga:Description>pathloss</pga:Description>
            </pga:UnMasked>
            <pga:Masked>
              <pga:SectionNumber>0</pga:SectionNumber>
              <pga:DataType>RasterSectionType_UnsignedChar</pga:DataType>
              <pga:Description>pathloss</pga:Description>
            </pga:Masked>
          </pga:Section>
          <pga:Section id="0">
            <pga:UnMasked>
              <pga:SectionNumber>1</pga:SectionNumber>
              <pga:DataType>RasterSectionType_Short</pga:DataType>
              <pga:Description>vertical masking angle</pga:Description>
            </pga:UnMasked>
            <pga:Masked xsi:nil="true" />
          </pga:Section>
        </pga:StandardPredRasterSections>
        <pga:WellKnownExtendedRasterSections>
          <pga:Section id="0">
            <pga:UnMasked>
              <pga:SectionNumber>200</pga:SectionNumber>
              <pga:DataType>RasterSectionType_UnsignedChar</pga:DataType>
              <pga:Description>line of sight state</pga:Description>
            </pga:UnMasked>
            <pga:Masked>
              <pga:SectionNumber>100</pga:SectionNumber>
              <pga:DataType>RasterSectionType_UnsignedChar</pga:DataType>
              <pga:Description>line of sight state</pga:Description>
```

8 Example XML Files

This section contains example XML files, which are used by the PGA loader for batch submissions using multiple or single files. See Submit Command in Section 7.1.

8.1 Example Antennas File (Passive)

This shows an example antenna file for an antenna type set to ‘Passive’:

```
<?xml version="1.0"?>
<ArrayOfAntenna xmlns:pga="http://www.teoco.com/pga/contracts/2017/03">
  <Antenna id="CMA_BDHH_6520_E0_8\CMA_BDHH_6520_E0_8_01D">
    <pga:AzimuthOffset>20</pga:AzimuthOffset>
    <pga:TiltType>Electrical</pga:TiltType>
    <pga:Tilt>2.0</pga:Tilt>
    <pga:GainType>dBI</pga:GainType>
    <pga:Gain>1.0</pga:Gain>
    <pga:Frequency>25.0</pga:Frequency>
    <pga:FrontToBackRatio>2.0</pga:FrontToBackRatio>
    <pga:CrossPolarDiscrimination>1.5</pga:CrossPolarDiscrimination>
    <pga:PolarisationType>Horizontal</pga:PolarisationType>
    <pga:HorizontalMask>{0,0;1,0;2,0.01;3,0.02}</pga:HorizontalMask>
    <pga:VerticalMask>{0,0;1,0;2,0.01;3,0.02}</pga:VerticalMask>
    <pga:PatternBeamIndex>-1</pga:PatternBeamIndex>
  </Antenna>
</ArrayOfAntenna>
```

This corresponds to the ‘Antenna’ type in the PGA web service description: see Section 4.4.1.

8.2 Example Antennas File (Switched Beam)

This shows an example antenna file for an antenna type set to ‘Switched Beam’:

```
<?xml version="1.0">
<ArrayOfSwitchedBeamAntenna xmlns:pga="http://www.teoco.com/pga/contracts/2017/03">
  <SwitchedBeamAntenna id="AIR5331_B260_TB_preliminary_V1.2">
    <pga:BeamSets>
      <pga:BeamSet id=”Set1”>
        <pga:BeamIndex index=”0” beamType=”Control” controlBeamIndex=”0”
trafficBeamIndex=”-1” />
        <pga:BeamIndex index=”1” beamType=”Traffic” controlBeamIndex=”-1”
trafficBeamIndex=”1” />
      </pga:BeamSet>
    </pga:BeamSets>
    <pga:Antennas>
      <pga:Antenna id="AIR5331_B260_TB_preliminary_V1.2_BI_0_f_37">
        <pga:AzimuthOffset>0.000000</pga:AzimuthOffset>
        <pga:TiltType>Mechanical</pga:TiltType>
        <pga:Tilt>0.000000</pga:Tilt>
        <pga:GainType>dBI</pga:GainType>
        <pga:Gain>28.379999</pga:Gain>
        <pga:Frequency>37000.000000</pga:Frequency>
        <pga:FrontToBackRatio>0.000000</pga:FrontToBackRatio>
        <pga:CrossPolarDiscrimination>0.000000</pga:CrossPolarDiscrimination>
        <pga:PolarisationType>Vertical</pga:PolarisationType>
        <pga:HorizontalMask>0,0;1,0;2,0.01;3,0.02</pga:HorizontalMask>
        <pga:VerticalMask>0,0;1,0;2,0.01;3,0.02</pga:VerticalMask>
        <pga:PatternBeamIndex>0</pga:PatternBeamIndex>
      </pga:Antenna>
```

```
      <pga:Antenna id="AIR5331_B260_TB_preliminary_V1.2_BI_0_f_40">
        <pga:AzimuthOffset>0.000000</pga:AzimuthOffset>
        <pga:TiltType>Mechanical</pga:TiltType>
        <pga:Tilt>0.000000</pga:Tilt>
        <pga:GainType>dBI</pga:GainType>
        <pga:Gain>29.040001</pga:Gain>
        <pga:Frequency>40000.000000</pga:Frequency>
        <pga:FrontToBackRatio>0.000000</pga:FrontToBackRatio>
        <pga:CrossPolarDiscrimination>0.000000</pga:CrossPolarDiscrimination>
        <pga:PolarisationType>Vertical</pga:PolarisationType>
        <pga:HorizontalMask>0,0;1,0;2,0.01;3,0.02</pga:HorizontalMask>
        <pga:VerticalMask>0,0;1,0;2,0.01;3,0.02</pga:VerticalMask>
        <pga:PatternBeamIndex>0</pga:PatternBeamIndex>
      </pga:Antenna>
      <pga:Antenna id="AIR5331_B260_TB_preliminary_V1.2_BI_1_f_37">
        <pga:AzimuthOffset>-58.000000</pga:AzimuthOffset>
        <pga:TiltType>Mechanical</pga:TiltType>
        <pga:Tilt>10.000000</pga:Tilt>
        <pga:GainType>dBI</pga:GainType>
        <pga:Gain>25.150000</pga:Gain>
        <pga:Frequency>37000.000000</pga:Frequency>
        <pga:FrontToBackRatio>0.000000</pga:FrontToBackRatio>
        <pga:CrossPolarDiscrimination>0.000000</pga:CrossPolarDiscrimination>
        <pga:PolarisationType>Vertical</pga:PolarisationType>
        <pga:HorizontalMask>0,0;1,0;2,0.01;3,0.02</pga:HorizontalMask>
        <pga:VerticalMask>0,0;1,0;2,0.01;3,0.02</pga:VerticalMask>
        <pga:PatternBeamIndex>1</pga:PatternBeamIndex>
      </pga:Antenna>
      <pga:Antenna id="AIR5331_B260_TB_preliminary_V1.2_BI_1_f_40">
        <pga:AzimuthOffset>-58.000000</pga:AzimuthOffset>
        <pga:TiltType>Mechanical</pga:TiltType>
        <pga:Tilt>10.000000</pga:Tilt>
        <pga:GainType>dBI</pga:GainType>
        <pga:Gain>25.420000</pga:Gain>
        <pga:Frequency>40000.000000</pga:Frequency>
        <pga:FrontToBackRatio>0.000000</pga:FrontToBackRatio>
        <pga:CrossPolarDiscrimination>0.000000</pga:CrossPolarDiscrimination>
        <pga:PolarisationType>Vertical</pga:PolarisationType>
        <pga:HorizontalMask>0,0;1,0;2,0.01;3,0.02</pga:HorizontalMask>
        <pga:VerticalMask>0,0;1,0;2,0.01;3,0.02</pga:VerticalMask>
        <pga:PatternBeamIndex>1</pga:PatternBeamIndex>
      </pga:Antenna>
    </pga:Antennas>
  </SwitchedBeamAntenna>
</ArrayOfSwitchedBeamAntenna>
```

8.3 Example Models File

This shows an example prediction model file:

```
<?xml version="1.0"?>
<ArrayOfPredictionModel xmlns:pga="http://www.teoco.com/pga/contracts/2017/03">
  <PredictionModel id="UTM11_Spokane_2100_v4_11">
    <pga:ClassId>{520E6383-8177-4F47-9F7E-E9AF64E44ABE}</pga:ClassId>
    <pga:Params>
      <MODEL-PARAMS VERSION="1">
        <!-- Content defined by selected propagation model -->
      </MODEL-PARAMS>
    </pga:Params>
  </PredictionModel>
</ArrayOfPredictionModel>
```

This corresponds to the ‘PredictionModel’ type in the PGA web service description: see Section 4.4.5.

8.4 Example Map Data Folder File

This shows an example map data folder file:

```
<?xml version="1.0"?>
<ArrayOfMapDataSet xmlns:pga="http://www.teoco.com/pga/contracts/2017/03">
  <MapDataSet moniker="Map01" epsg="32630">
    <pga:ClutterFolder>\\Source\Clutter\Data</pga:ClutterFolder>
    <pga:DTMFolder>\\Source\Height\Data</pga:DTMFolder>
    <pga:BuildingRasterFolder>\\Source\BuildingRaster\Data</pga:BuildingRasterFolder>
    <pga:BuildingVectorFolder>\\Source\BuildingVector\Data</pga:BuildingVectorFolder>
  </MapDataSet>
</ArrayOfMapDataSet>
```

This corresponds to the ‘MapDataSet’ type in the PGA web service description: see Section 4.4.6.

Important: If you have enabled distribution on the PGA service, the paths for clutter and height data must be in UNC format.

8.5 Example Output Folder File

This shows an example output folder file:

```
<?xml version="1.0"?>
<ArrayOfOutputFolder>
  <OutputFolder moniker="PredFolder1">\\dest\pred\output1</OutputFolder>
  <OutputFolder moniker="PredFolder2">\\dest\pred\output2</OutputFolder>
</ArrayOfOutputFolder>
```

This corresponds to the ‘OutputFolder’ type in the PGA web service description: see Section 4.4.7.

Important: If you have enabled distribution on the PGA service, the paths for clutter and height data must be in UNC format.

8.6 Example Jobs File

This shows an example jobs file:

```
<?xml version="1.0"?>
<ArrayOfPredictionJobItem xmlns:pga="http://www.teoco.com/pga/contracts/2017/03">
  <PredictionJobItem>
    <pga:ItemID>0</pga:ItemID>
    <pga:Location epsg="4269">
      <pga:X>-117.8695</pga:X>
      <pga:Y>48.914802</pga:Y>
      <pga:Z>0</pga:Z>
    </pga:Location>
    <pga:Output mapInputMoniker="Map01" predOutputMoniker="PredFolder1" />
    <pga:AntennaParameters id="CMA_BDHH_6520_E0_8\CMA_BDHH_6520_E0_8_01D">
      <pga:Azimuth_Degrees>0</pga:Azimuth_Degrees>
      <pga:MechanicalTilt_Degrees>0</pga:MechanicalTilt_Degrees>
      <pga:Height_Cm>3048</pga:Height_Cm>
    </pga:AntennaParameters>
    <pga:PredictionModel id="UTM11_Spokane_2100_v4_11">
      <pga:Radius_Cm>3500000</pga:Radius_Cm>
      <pga:Resolution_Cm>2500</pga:Resolution_Cm>
    </pga:PredictionModel>
  </PredictionJobItem>
</ArrayOfPredictionJobItem>
```

This corresponds to the ‘PredictionJobItem’ type in the PGA web service description: see Section 4.4.8.

8.7 Example Batch File

The content shown below is a combined version of the examples above, which can be used in single file batch submissions:

```
<?xml version="1.0"?>
<BatchJobItem xmlns:pga="http://www.teoco.com/pga/contracts/2017/03">
  <ArrayOfAntenna>
    <Antenna id="CMA_BDHH_6520_E0_8\CMA_BDHH_6520_E0_8_01D">
      <pga:AzimuthOffset>20</pga:AzimuthOffset>
      <pga:TiltType>Electrical</pga:TiltType>
      <pga:Tilt>2.0</pga:Tilt>
      <pga:GainType>dBI</pga:GainType>
      <pga:Gain>1.0</pga:Gain>
      <pga:Frequency>25.0</pga:Frequency>
      <pga:FrontToBackRatio>2.0</pga:FrontToBackRatio>
      <pga:CrossPolarDiscrimination>1.5</pga:CrossPolarDiscrimination>
      <pga:PolarisationType>Horizontal</pga:PolarisationType>
      <pga:HorizontalMask>{0,0;1,0;2,0.01;3,0.02}</pga:HorizontalMask>
      <pga:VerticalMask>{0,0;1,0;2,0.01;3,0.02}</pga:VerticalMask>
    </Antenna>
  </ArrayOfAntenna>
  <ArrayOfSwitchedBeamAntenna>
    <SwitchedBeamAntenna id="AIR5331_B260_TB_preliminary_V1.2">
      <pga:BeamSets>
        <pga:BeamSet id=”Set1”>
          <pga:BeamIndex index=”0” beamType=”Control” controlBeamIndex=”0”
trafficBeamIndex=”-1” />
          <pga:BeamIndex index=”1” beamType=”Traffic” controlBeamIndex=”-1”
trafficBeamIndex=”1” />
        </pga:BeamSet>
      </pga:BeamSets>
      <pga:Antennas>
        <pga:Antenna id="AIR5331_B260_TB_preliminary_V1.2_BI_0_f_37">
          <pga:AzimuthOffset>0.000000</pga:AzimuthOffset>
          <pga:TiltType>Mechanical</pga:TiltType>
          <pga:Tilt>0.000000</pga:Tilt>
          <pga:GainType>dBI</pga:GainType>
          <pga:Gain>28.379999</pga:Gain>
          <pga:Frequency>37000.000000</pga:Frequency>
          <pga:FrontToBackRatio>0.000000</pga:FrontToBackRatio>
          <pga:CrossPolarDiscrimination>0.000000</pga:CrossPolarDiscrimination>
          <pga:PolarisationType>Vertical</pga:PolarisationType>
          <pga:HorizontalMask>0,0;1,0;2,0.01;3,0.02</pga:HorizontalMask>
          <pga:VerticalMask>0,0;1,0;2,0.01;3,0.02</pga:VerticalMask>
          <pga:PatternBeamIndex>0</pga:PatternBeamIndex>
        </pga:Antenna>
        <pga:Antenna id="AIR5331_B260_TB_preliminary_V1.2_BI_0_f_40">
          <pga:AzimuthOffset>0.000000</pga:AzimuthOffset>
          <pga:TiltType>Mechanical</pga:TiltType>
          <pga:Tilt>0.000000</pga:Tilt>
          <pga:GainType>dBI</pga:GainType>
          <pga:Gain>29.040001</pga:Gain>
          <pga:Frequency>40000.000000</pga:Frequency>
          <pga:FrontToBackRatio>0.000000</pga:FrontToBackRatio>
          <pga:CrossPolarDiscrimination>0.000000</pga:CrossPolarDiscrimination>
          <pga:PolarisationType>Vertical</pga:PolarisationType>
          <pga:HorizontalMask>0,0;1,0;2,0.01;3,0.02</pga:HorizontalMask>
          <pga:VerticalMask>0,0;1,0;2,0.01;3,0.02</pga:VerticalMask>
          <pga:PatternBeamIndex>0</pga:PatternBeamIndex>
        </pga:Antenna>
        <pga:Antenna id="AIR5331_B260_TB_preliminary_V1.2_BI_1_f_37">
          <pga:AzimuthOffset>-58.000000</pga:AzimuthOffset>
          <pga:TiltType>Mechanical</pga:TiltType>
          <pga:Tilt>10.000000</pga:Tilt>
          <pga:GainType>dBI</pga:GainType>
          <pga:Gain>25.150000</pga:Gain>
```

```
        <pga:Frequency>37000.000000</pga:Frequency>
        <pga:FrontToBackRatio>0.000000</pga:FrontToBackRatio>
        <pga:CrossPolarDiscrimination>0.000000</pga:CrossPolarDiscrimination>
        <pga:PolarisationType>Vertical</pga:PolarisationType>
        <pga:HorizontalMask>0,0;1,0;2,0.01;3,0.02</pga:HorizontalMask>
        <pga:VerticalMask>0,0;1,0;2,0.01;3,0.02</pga:VerticalMask>
        <pga:PatternBeamIndex>1</pga:PatternBeamIndex>
      </pga:Antenna>
      <pga:Antenna id="AIR5331_B260_TB_preliminary_V1.2_BI_1_f_40">
        <pga:AzimuthOffset>-58.000000</pga:AzimuthOffset>
        <pga:TiltType>Mechanical</pga:TiltType>
        <pga:Tilt>10.000000</pga:Tilt>
        <pga:GainType>dBI</pga:GainType>
        <pga:Gain>25.420000</pga:Gain>
        <pga:Frequency>40000.000000</pga:Frequency>
        <pga:FrontToBackRatio>0.000000</pga:FrontToBackRatio>
        <pga:CrossPolarDiscrimination>0.000000</pga:CrossPolarDiscrimination>
        <pga:PolarisationType>Vertical</pga:PolarisationType>
        <pga:HorizontalMask>0,0;1,0;2,0.01;3,0.02</pga:HorizontalMask>
        <pga:VerticalMask>0,0;1,0;2,0.01;3,0.02</pga:VerticalMask>
        <pga:PatternBeamIndex>1</pga:PatternBeamIndex>
      </pga:Antenna>
    </pga:Antennas>
  </SwitchedBeamAntenna>
</ArrayOfSwitchedBeamAntenna>
<ArrayOfPredictionModel>
  <PredictionModel id="UTM11_Spokane_2100_v4_11">
    <pga:ClassId>{520E6383-8177-4F47-9F7E-E9AF64E44ABE}</pga:ClassId>
    <pga:Params>
      <MODEL-PARAMS VERSION="1">
        <!-- Content defined by selected propagation model -->
      </MODEL-PARAMS>
    </pga:Params>
  </PredictionModel>
</ArrayOfPredictionModel>
<ArrayOfMapDataSet>
  <MapDataSet moniker="Map01" epsg="32630">
    <pga:ClutterFolder>\\Source\Clutter\Data</pga:ClutterFolder>
    <pga:DTMFolder>\\Source\Height\Data</pga:DTMFolder>
    <pga:BuildingRasterFolder>\\Source\BuildRaster\Data</pga:BuildingRasterFolder>
    <pga:BuildingVectorFolder>\\Source\BuildVector\Data</pga:BuildingVectorFolder>
  </MapDataSet>
</ArrayOfMapDataSet>
<ArrayOfOutputFolder>
  <OutputFolder moniker="PredFolder1">\\dest\pred\output1</OutputFolder>
  <OutputFolder moniker="PredFolder2">\\dest\pred\output2</OutputFolder>
</ArrayOfOutputFolder>
<ArrayOfPredictionJobItem>
  <PredictionJobItem>
    <pga:ItemID>0</pga:ItemID>
    <pga:Location epsg="4269">
      <pga:X>-117.8695</pga:X>
      <pga:Y>48.914802</pga:Y>
      <pga:Z>0</pga:Z>
    </pga:Location>
    <pga:Output mapInputMoniker="Map01" predOutputMoniker="PredFolder1" />
    <pga:AntennaParameters id="CMA_BDHH_6520_E0_8\CMA_BDHH_6520_E0_8_01D">
      <pga:Azimuth_Degrees>0</pga:Azimuth_Degrees>
      <pga:MechanicalTilt_Degrees>0</pga:MechanicalTilt_Degrees>
      <pga:Height_Cm>3048</pga:Height_Cm>
    </pga:AntennaParameters>
    <pga:PredictionModel id="UTM11_Spokane_2100_v4_11">
     <pga:Radius_Cm>3500000</pga:Radius_Cm>
     <pga:Resolution_Cm>2500</pga:Resolution_Cm>
    </pga:PredictionModel>
  </PredictionJobItem>
  <PredictionJobItem>
    <pga:ItemID>1</pga:ItemID>
    <pga:Location epsg="4269">
      <pga:X>-114.1264</pga:X>
      <pga:Y>46.16332</pga:Y>
      <pga:Z>0</pga:Z>
```

```
      </pga:Location>
      <pga:Output mapInputMoniker="Map01" predOutputMoniker="PredFolder1" />
      <pga:AntennaParameters id="AIR5331_B260_TB_preliminary_V1.2">
        <pga:Azimuth_Degrees>0</pga:Azimuth_Degrees>
        <pga:MechanicalTilt_Degrees>0</pga:MechanicalTilt_Degrees>
        <pga:Height_Cm>3048</pga:Height_Cm>
        <pga:SwitchedBeam_Frequency>37000</pga:SwitchedBeam_Frequency>
        <pga:SwitchedBeam_BeamSetMoniker>Set1</pga:SwitchedBeam_BeamSetMoniker>
      </pga:AntennaParameters>
      <pga:PredictionModel id="UTM11_Spokane_2100_v4_11">
       <pga:Radius_Cm>3500000</pga:Radius_Cm>
       <pga:Resolution_Cm>2500</pga:Resolution_Cm>
      </pga:PredictionModel>
    </PredictionJobItem>
  </ArrayOfPredictionJobItem>
  <pga:Options>AllowHeightSmoothing AllowPlcCorrection</pga:Options>
</BatchJobItem>
```

Important: If you have enabled distribution on the PGA service, the paths for clutter data, height data and output must be in UNC format.

9 Troubleshooting the PGA

This chapter describes some of the possible sources of help and information when you experience problems with the PGA.

9.1 Logging in the PGA

By default, the PGA service logs all errors and exceptions to the ‘pga_log.xml’ file, located at ‘Program Files\TEOCO\ENTERPRISE\Common’.

However, if you want to log other types of data in the pga_log:

1. Open the ‘logging.config’ file, located at ‘Program Files\TEOCO\ENTERPRISE’.
2. Navigate to the <categorySources> section.

   This section describes the different information ‘switches’ and to which ‘listener(s)’ they are sent:

```
<categorySources>
  <!-- Uncomment below to trace soap envelopes paseed into PGA -->
  <!--<add switchValue="All" name="WCFDebugMessageInspector">
    <listeners>
      <add name="Console Listener" />
    </listeners>
  </add>-->
  <add switchValue="Error" name="Exceptions">
    <listeners>
      <add name="Windows Event Log Errors TraceListener" />
      <add name="XML File Listener" />
    </listeners>
  </add>
  <add switchValue="Error" name="Errors">
    <listeners>
      <add name="Console Listener" />
      <add name="Windows Event Log Errors TraceListener" />
      <add name="XML File Listener" />
    </listeners>
  </add>
  <add switchValue="Information" name="Event">
    <listeners>
      <add name="Console Listener" />
    </listeners>
  </add>
  <add switchValue="All" name="Information">
    <listeners>
      <add name="Console Listener" />
    </listeners>
  </add>
  <add switchValue="All" name="Messaging">
    <listeners>
      <add name="Console Listener" />
    </listeners>
  </add>
  <add switchValue="All" name="Status">
    <listeners>
      <add name="Console Listener" />
    </listeners>
  </add>
  <add switchValue="All" name="System.ServiceModel" />
  <add switchValue="ActivityTracing" name="Trace" />
  <add switchValue="Warning" name="Warnings" />
</categorySources>
```

   In addition to errors and exceptions, you can also log data related to events (‘Event’), status (‘Status’), messaging (‘Messaging’) and other information (‘Information’).

4. To display a particular information type in the pga_log, ensure that the list of listeners contains the ‘XML File Listener’:

```
<add switchValue="Error" name="Exceptions">
  <listeners>
    <add name="Windows Event Log Errors TraceListener" />
    <add name="XML File Listener" />
  </listeners>
```

5. If you do not want to display a particular information type in the pga_log, ensure that the ‘XML Listener’ is not included for that type.

9.2 Troubleshooting with the Event Viewer

If you experience problems with the PGA installation, the first place to check is the Event Viewer. It logs all errors, and you can use it to find possible reasons why the installation was unsuccessful.

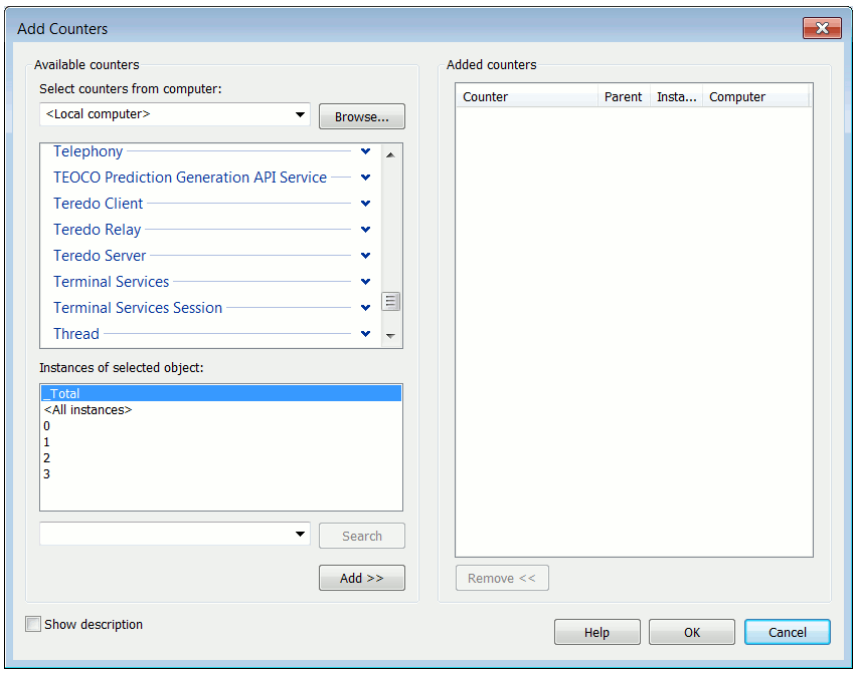
9.3 Troubleshooting with the Performance Monitor

You can use the Windows Performance Monitor to view performance counters for the web services. To set up the Performance Monitor:

1. Click the **Start** menu, click **Run** and type **perfmon.msc**.
2. From the tree, expand the **Monitoring Tools** folder, and then click **Performance Monitor**.
3. In the right-hand pane, click the **Add** button.

   - or -

   Use the keyboard shortcut **CTRL+N**.



4. Expand the **TEOCO Prediction Generation API Service**, and select the counters you want to monitor (for example, ‘Total number of jobs cancelled’).
5. Click the **Add>>** button.
6. Click **OK** to close the screen after you have added the required counters.