Autoreduction setup forms

Mathieu Doucet

Oak Ridge National Laboratory

Plan for the next few weeks

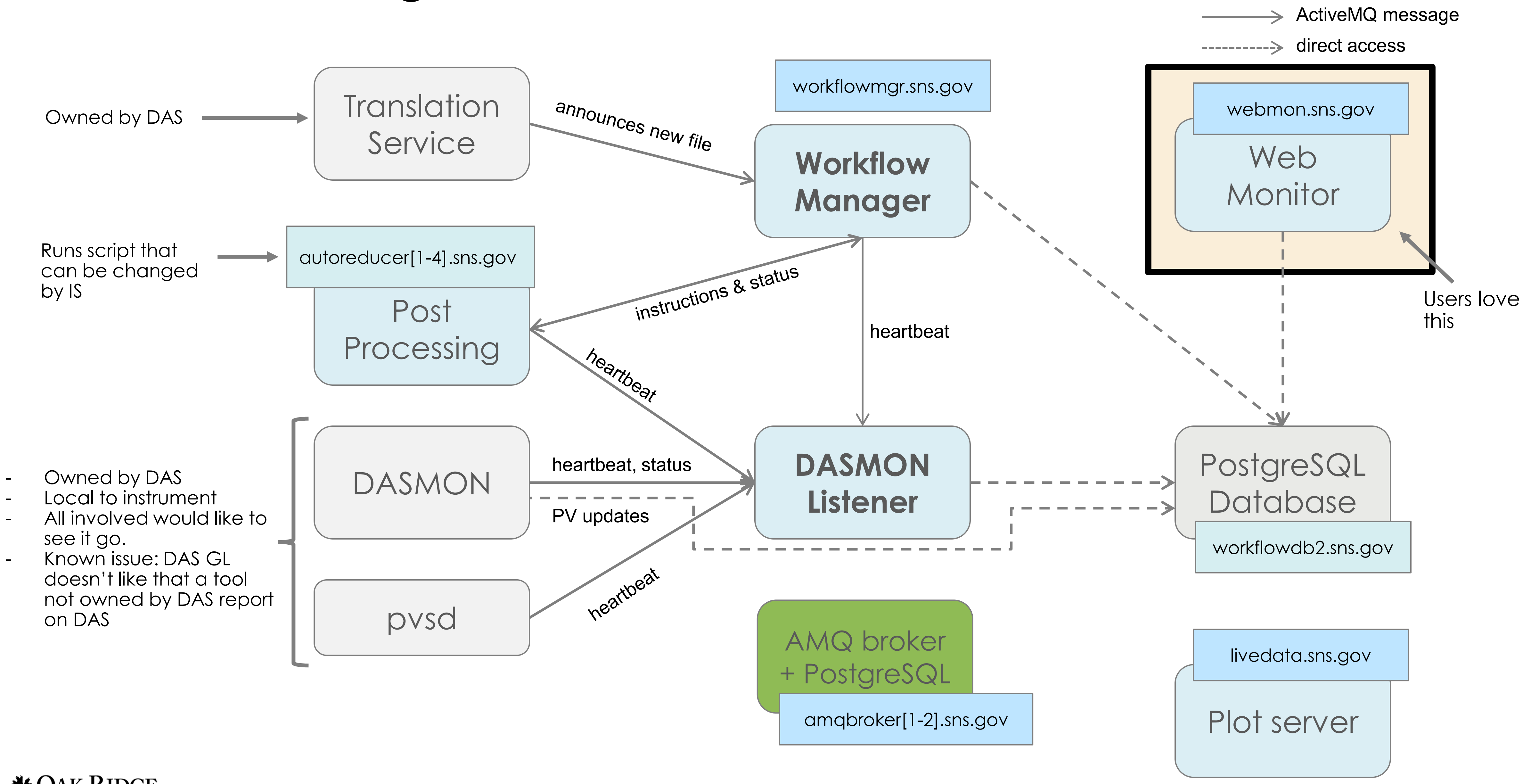
Test environment:

New RHEL8 machines are being set up so we can install them together

Topics to cover:

1. General overview
2. Workflow manager and DASMON listener – Installation & maintenance [this presentation]
3. Web monitor – Installation and maintenance
4. Autoreduction service – Installation and maintenance
5. **Autoreduction setup through webmon**
6. The IHC call – when things go wrong & recovery strategies
7. Vision for the future – what I would do differently

Post-Processing Architecture



- Owned by DAS
- Local to instrument
- All involved would like to see it go.
- Known issue: DAS GL doesn't like that a tool not owned by DAS report on DAS

Setup accessible through web monitor

The code is in data_workflow/reporting/reduction

User submits form

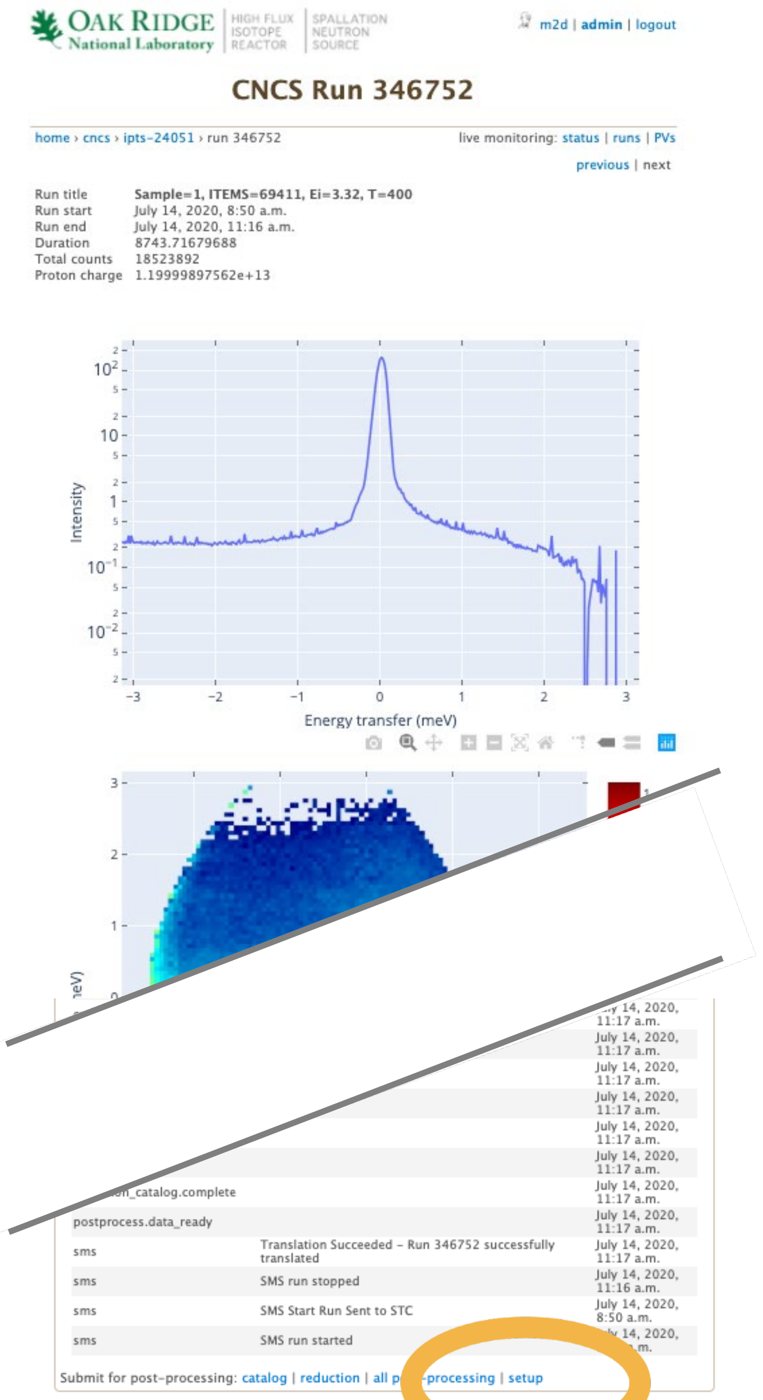
- Most recent parameter values are stored in the DB
- Form allows user to see/modify parameters

Process form

- Store new values in DB
- Compose json payload
- Submit request to AR through AMQ

Write script

- AR service receives request
- Template is used to generate script
- /SNS/CNCS/shared/autoreduce/reduce_CNCS.py.template


m2d | admin | logout

CNCS Run 346752

home › cncs › ipts-24051 › run 346752 live monitoring: status | runs | PVs

previous | next

Run title	**Sample=1, ITEMS=69411, Ei=3.32, T=400**
Run start	July 14, 2020, 8:50 a.m.
Run end	July 14, 2020, 11:16 a.m.
Duration	8743.71679688
Total counts	18523892
Proton charge	1.19999897562e+13

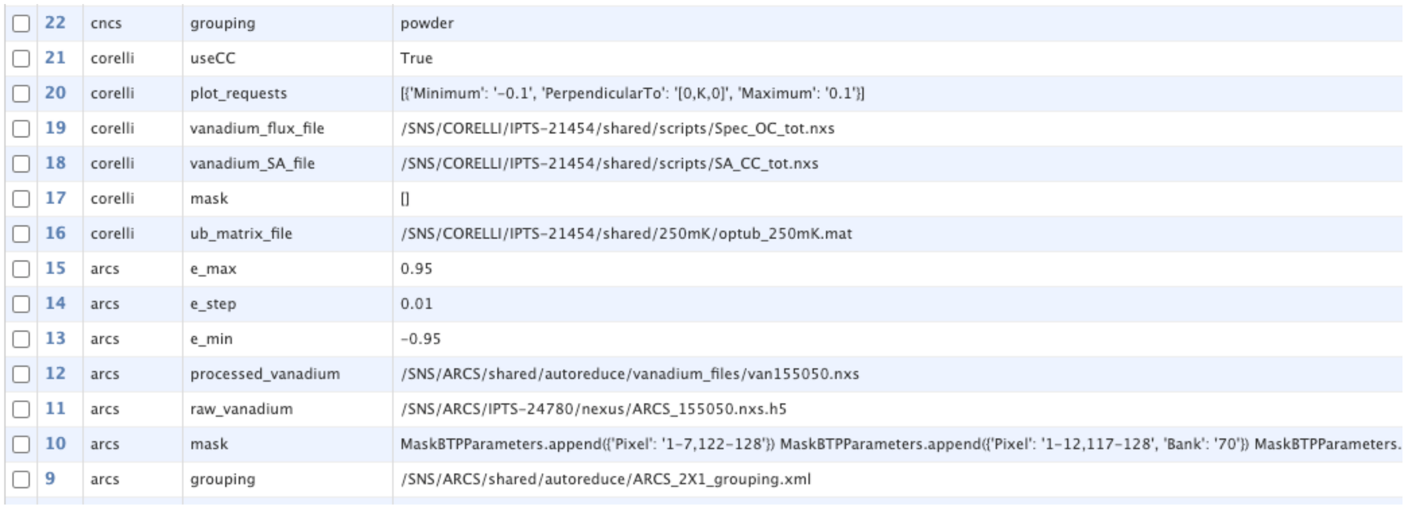
		July 14, 2020, 11:17 a.m.
		July 14, 2020, 11:17 a.m.
		July 14, 2020, 11:17 a.m.
		July 14, 2020, 11:17 a.m.
		July 14, 2020, 11:17 a.m.
		July 14, 2020, 11:17 a.m.
catalog.complete		July 14, 2020, 11:17 a.m.
postprocess.data_ready		July 14, 2020, 11:17 a.m.
sms	Translation Succeeded - Run 346752 successfully translated	July 14, 2020, 11:17 a.m.
sms	SMS run stopped	July 14, 2020, 11:16 a.m.
sms	SMS Start Run Sent to STC	July 14, 2020, 8:50 a.m.
sms	SMS run started	14, 2020, m.

Submit for post-processing: catalog | reduction | all p -processing | setup

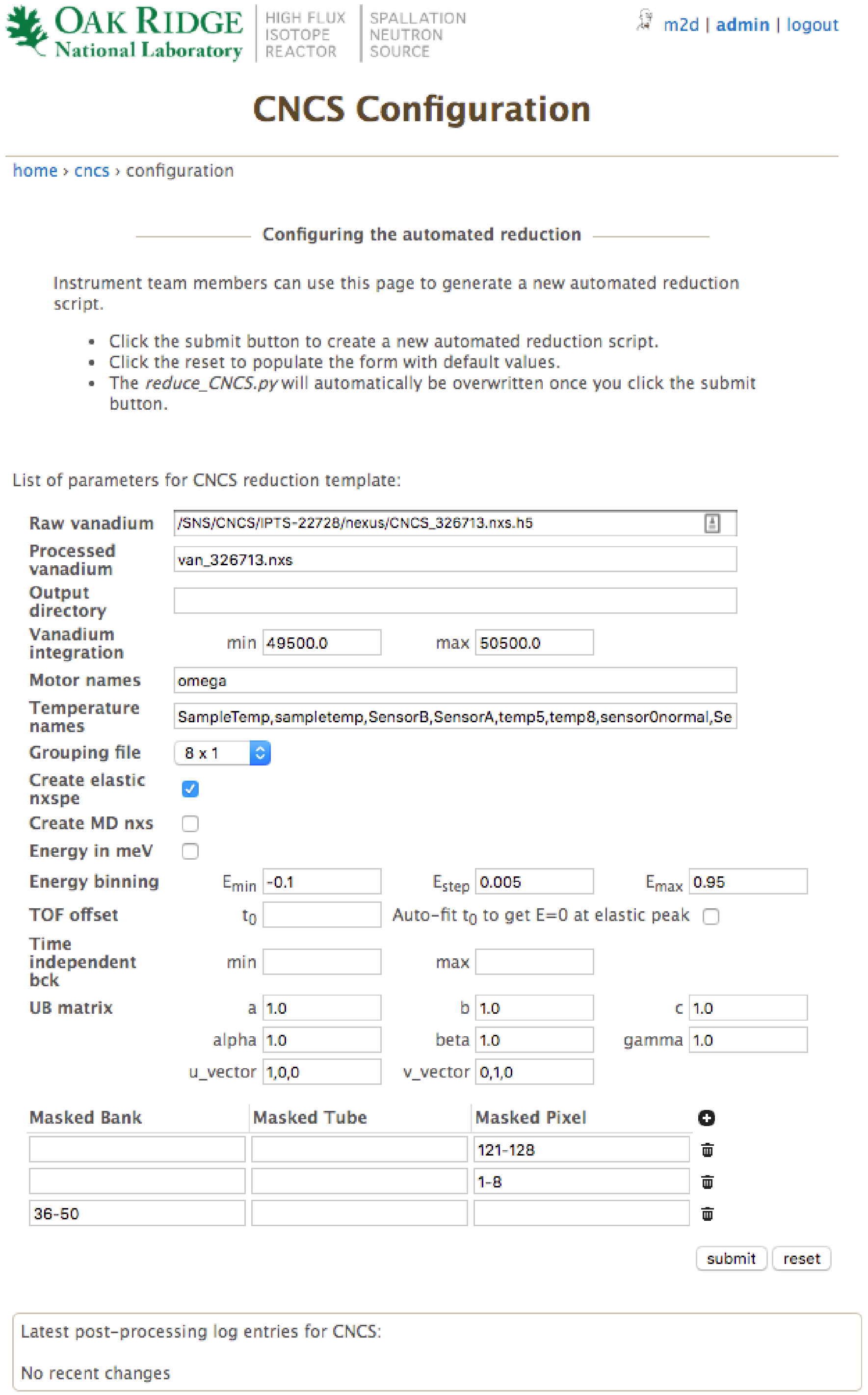
The setup form

https://monitor.sns.gov/reduction/cncs/

- Code is in data_workflow/reporting/reduction/forms.py
- Inherit from a common base class **BaseReductionConfigurationForm**
- Values are stored as ReductionProperty model objects
- New params can be added to the form and will be created in the DB automatically
- An html template also needs to be modified:
  - data_workflow/reporting/templates/reduction/configuration_cncs.html



☐	22	cncs	grouping	powder
☐	21	corelli	useCC	True
☐	20	corelli	plot_requests	[{'Minimum': '-0.1', 'PerpendicularTo': '[0,K,0]', 'Maximum': '0.1'}]
☐	19	corelli	vanadium_flux_file	/SNS/CORELLI/IPTS-21454/shared/scripts/Spec_OC_tot.nxs
☐	18	corelli	vanadium_SA_file	/SNS/CORELLI/IPTS-21454/shared/scripts/SA_CC_tot.nxs
☐	17	corelli	mask	[]
☐	16	corelli	ub_matrix_file	/SNS/CORELLI/IPTS-21454/shared/250mK/optub_250mK.mat
☐	15	arcs	e_max	0.95
☐	14	arcs	e_step	0.01
☐	13	arcs	e_min	-0.95
☐	12	arcs	processed_vanadium	/SNS/ARCS/shared/autoreduce/vanadium_files/van155050.nxs
☐	11	arcs	raw_vanadium	/SNS/ARCS/IPTS-24780/nexus/ARCS_155050.nxs.h5
☐	10	arcs	mask	MaskBTPParameters.append({'Pixel': '1-7,122-128'}) MaskBTPParameters.append({'Pixel': '1-12,117-128', 'Bank': '70'}) MaskBTPParameters.
☐	9	arcs	grouping	/SNS/ARCS/shared/autoreduce/ARCS_2X1_grouping.xml



OAK RIDGE National Laboratory | HIGH FLUX ISOTOPE REACTOR | SPALLATION NEUTRON SOURCE

m2d | admin | logout

CNCS Configuration

home › cncs › configuration

Configuring the automated reduction

Instrument team members can use this page to generate a new automated reduction script.

- Click the submit button to create a new automated reduction script.
- Click the reset to populate the form with default values.
- The *reduce_CNCS.py* will automatically be overwritten once you click the submit button.

List of parameters for CNCS reduction template:

Raw vanadium	/SNS/CNCS/IPTS-22728/nexus/CNCS_326713.nxs.h5					
Processed vanadium	van_326713.nxs					
Output directory						
Vanadium integration	min	49500.0	max	50500.0		
Motor names	omega					
Temperature names	SampleTemp,sampletemp,SensorB,SensorA,temp5,temp8,sensor0normal,Se					
Grouping file	8 x 1					
Create elastic nxspe	☑					
Create MD nxs	☐					
Energy in meV	☐					
Energy binning	E_{min}	-0.1	E_{step}	0.005	E_{max}	0.95
TOF offset	t_0		Auto-fit t_0 to get E=0 at elastic peak ☐			
Time independent bck	min		max			
UB matrix	a	1.0	b	1.0	c	1.0
	alpha	1.0	beta	1.0	gamma	1.0
	u_vector	1,0,0	v_vector	0,1,0		

Masked Bank	Masked Tube	Masked Pixel
		121-128
		1-8
36-50		

Latest post-processing log entries for CNCS:

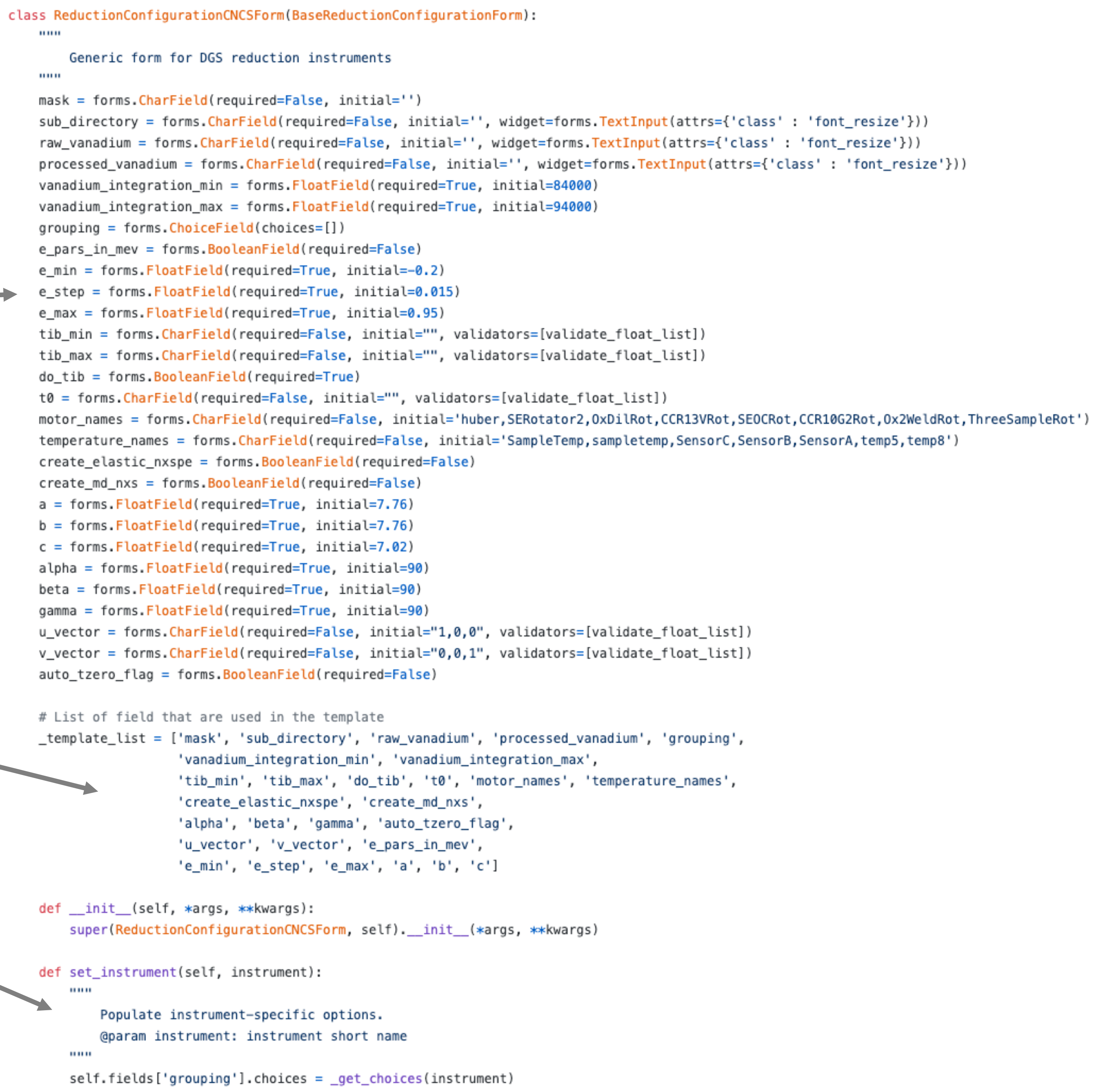
No recent changes

Example form

Uses standard Django fields

List of fields used in the template

If you need to populate drop-downs according to instrument, or initialize fields dynamically, do it here



```
class ReductionConfigurationCNCSForm(BaseReductionConfigurationForm):
    """
        Generic form for DGS reduction instruments
    """
    mask = forms.CharField(required=False, initial='')
    sub_directory = forms.CharField(required=False, initial='', widget=forms.TextInput(attrs={'class' : 'font_resize'}))
    raw_vanadium = forms.CharField(required=False, initial='', widget=forms.TextInput(attrs={'class' : 'font_resize'}))
    processed_vanadium = forms.CharField(required=False, initial='', widget=forms.TextInput(attrs={'class' : 'font_resize'}))
    vanadium_integration_min = forms.FloatField(required=True, initial=84000)
    vanadium_integration_max = forms.FloatField(required=True, initial=94000)
    grouping = forms.ChoiceField(choices=[])
    e_pars_in_mev = forms.BooleanField(required=False)
    e_min = forms.FloatField(required=True, initial=-0.2)
    e_step = forms.FloatField(required=True, initial=0.015)
    e_max = forms.FloatField(required=True, initial=0.95)
    tib_min = forms.CharField(required=False, initial="", validators=[validate_float_list])
    tib_max = forms.CharField(required=False, initial="", validators=[validate_float_list])
    do_tib = forms.BooleanField(required=True)
    t0 = forms.CharField(required=False, initial="", validators=[validate_float_list])
    motor_names = forms.CharField(required=False, initial='huber,SERotator2,OxDilRot,CCR13VRot,SEOCRot,CCR10G2Rot,Ox2WeldRot,ThreeSampleRot')
    temperature_names = forms.CharField(required=False, initial='SampleTemp,sampletemp,SensorC,SensorB,SensorA,temp5,temp8')
    create_elastic_nxspe = forms.BooleanField(required=False)
    create_md_nxs = forms.BooleanField(required=False)
    a = forms.FloatField(required=True, initial=7.76)
    b = forms.FloatField(required=True, initial=7.76)
    c = forms.FloatField(required=True, initial=7.02)
    alpha = forms.FloatField(required=True, initial=90)
    beta = forms.FloatField(required=True, initial=90)
    gamma = forms.FloatField(required=True, initial=90)
    u_vector = forms.CharField(required=False, initial="1,0,0", validators=[validate_float_list])
    v_vector = forms.CharField(required=False, initial="0,0,1", validators=[validate_float_list])
    auto_tzero_flag = forms.BooleanField(required=False)

    # List of field that are used in the template
    _template_list = ['mask', 'sub_directory', 'raw_vanadium', 'processed_vanadium', 'grouping',
                      'vanadium_integration_min', 'vanadium_integration_max',
                      'tib_min', 'tib_max', 'do_tib', 't0', 'motor_names', 'temperature_names',
                      'create_elastic_nxspe', 'create_md_nxs',
                      'alpha', 'beta', 'gamma', 'auto_tzero_flag',
                      'u_vector', 'v_vector', 'e_pars_in_mev',
                      'e_min', 'e_step', 'e_max', 'a', 'b', 'c']

    def __init__(self, *args, **kwargs):
        super(ReductionConfigurationCNCSForm, self).__init__(*args, **kwargs)

    def set_instrument(self, instrument):
        """
            Populate instrument-specific options.
            @param instrument: instrument short name
        """
        self.fields['grouping'].choices = _get_choices(instrument)
```

OAK RIDGE
National Laboratory

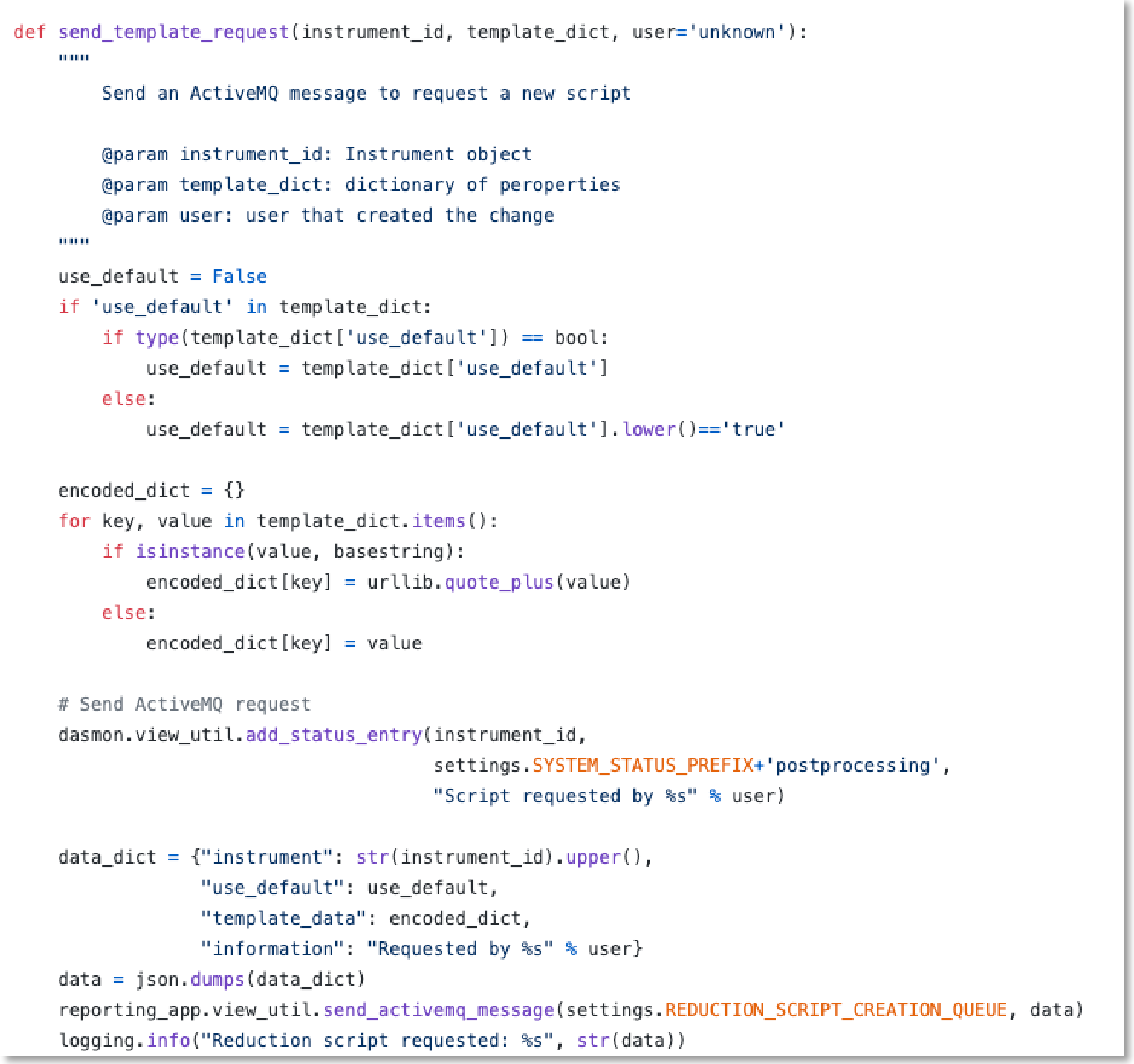
The AMQ message

- The code executing the job is part of the post_processing_agent:

  postprocessing/reduction_script_writer.py

- The web monitor doesn't have access to the file system, but the AR service does…

data_workflow/reporting/reduction/view_utils.py

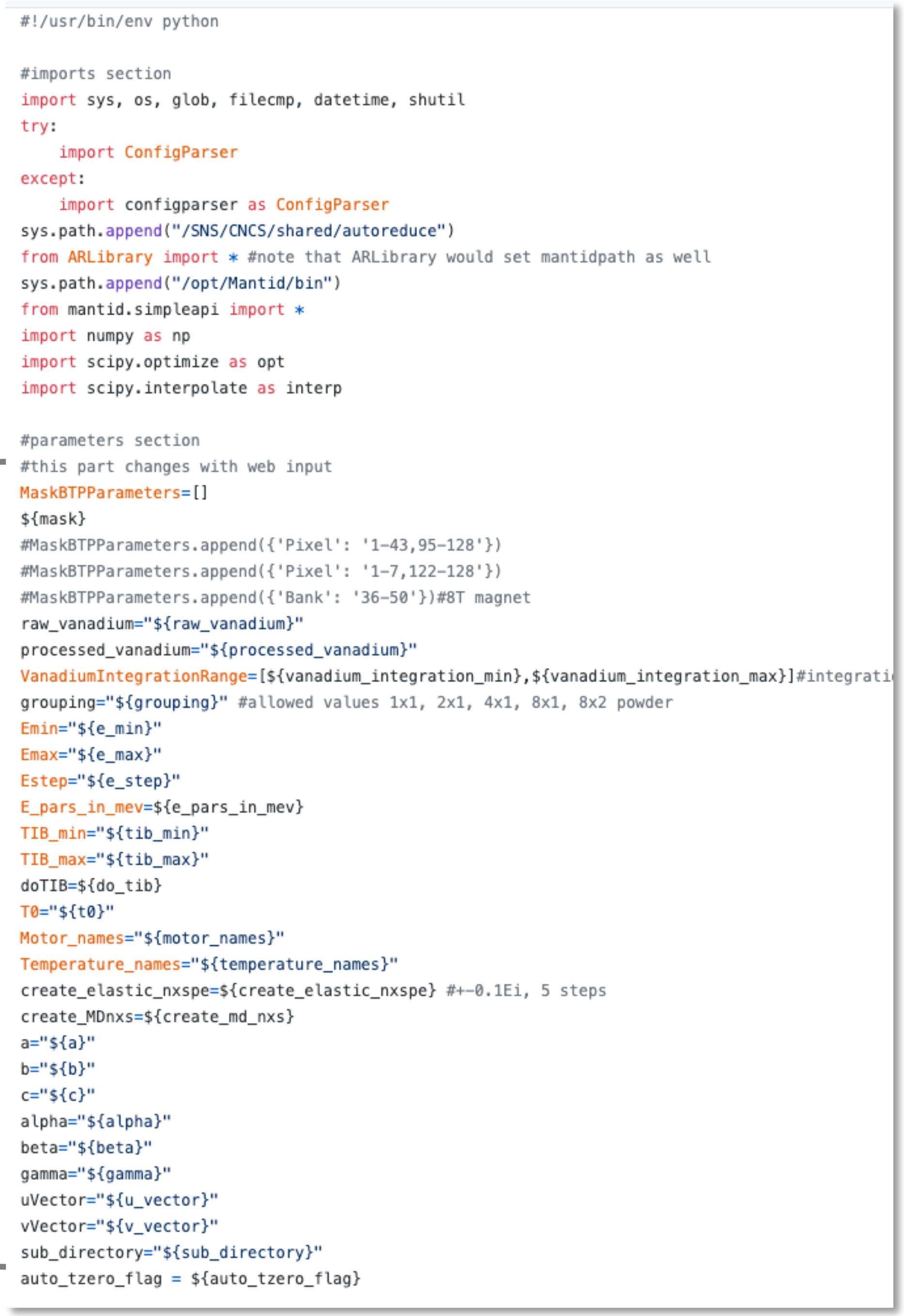


```
def send_template_request(instrument_id, template_dict, user='unknown'):
    """
        Send an ActiveMQ message to request a new script

        @param instrument_id: Instrument object
        @param template_dict: dictionary of peroperties
        @param user: user that created the change
    """
    use_default = False
    if 'use_default' in template_dict:
        if type(template_dict['use_default']) == bool:
            use_default = template_dict['use_default']
        else:
            use_default = template_dict['use_default'].lower()=='true'

    encoded_dict = {}
    for key, value in template_dict.items():
        if isinstance(value, basestring):
            encoded_dict[key] = urllib.quote_plus(value)
        else:
            encoded_dict[key] = value

    # Send ActiveMQ request
    dasmon.view_util.add_status_entry(instrument_id,
                                      settings.SYSTEM_STATUS_PREFIX+'postprocessing',
                                      "Script requested by %s" % user)

    data_dict = {"instrument": str(instrument_id).upper(),
                 "use_default": use_default,
                 "template_data": encoded_dict,
                 "information": "Requested by %s" % user}
    data = json.dumps(data_dict)
    reporting_app.view_util.send_activemq_message(settings.REDUCTION_SCRIPT_CREATION_QUEUE, data)
    logging.info("Reduction script requested: %s", str(data))
```

The template

- Template resides in the instrument's shared area
- It can be modified by the instrument team

Replace keys with values



```
#!/usr/bin/env python

#imports section
import sys, os, glob, filecmp, datetime, shutil
try:
    import ConfigParser
except:
    import configparser as ConfigParser
sys.path.append("/SNS/CNCS/shared/autoreduce")
from ARLibrary import * #note that ARLibrary would set mantidpath as well
sys.path.append("/opt/Mantid/bin")
from mantid.simpleapi import *
import numpy as np
import scipy.optimize as opt
import scipy.interpolate as interp

#parameters section
#this part changes with web input
MaskBTPParameters=[]
${mask}
#MaskBTPParameters.append({'Pixel': '1-43,95-128'})
#MaskBTPParameters.append({'Pixel': '1-7,122-128'})
#MaskBTPParameters.append({'Bank': '36-50'})#8T magnet
raw_vanadium="${raw_vanadium}"
processed_vanadium="${processed_vanadium}"
VanadiumIntegrationRange=[${vanadium_integration_min},${vanadium_integration_max}]#integrati
grouping="${grouping}" #allowed values 1x1, 2x1, 4x1, 8x1, 8x2 powder
Emin="${e_min}"
Emax="${e_max}"
Estep="${e_step}"
E_pars_in_mev=${e_pars_in_mev}
TIB_min="${tib_min}"
TIB_max="${tib_max}"
doTIB=${do_tib}
T0="${t0}"
Motor_names="${motor_names}"
Temperature_names="${temperature_names}"
create_elastic_nxspe=${create_elastic_nxspe} #+-0.1Ei, 5 steps
create_MDnxs=${create_md_nxs}
a="${a}"
b="${b}"
c="${c}"
alpha="${alpha}"
beta="${beta}"
gamma="${gamma}"
uVector="${u_vector}"
vVector="${v_vector}"
sub_directory="${sub_directory}"
auto_tzero_flag = ${auto_tzero_flag}
```