



Import Manager Playbook

Introduction

This playbook will guide prospective institutions through a setup project for MPOG. It describes the overall process as well as providing step-by-step instructions for each phase.

What is MPOG?

MPOG is a consortium of over 50 medical centers contributing electronic health record (EHR) and administrative data for the purposes of research and quality improvement.

What is ASPIRE?

The Anesthesiology Performance Improvement and Reporting Exchange (ASPIRE) is the quality improvement arm of MPOG and has developed several anesthesia-specific quality measures that are used to provide monthly feedback to anesthesiologists and CRNAs. This performance data can also be used for participation in the Merit-Based Incentive Payment (MIPS) program as ASPIRE is designated as a Qualified Clinical Data Registry (QCDR) by the Centers for Medicaid and Medicare Services (CMS). Organizations who are submitting data to MPOG can opt to participate with ASPIRE and/or the QCDR program.

How to Join?

There are four main stages for joining as a contributing institution in MPOG. These stages are laid out in detail later in this playbook, but to summarize they are:

1. Laying the Groundwork

In the beginning phase of the project, the main focus is on getting organized by reviewing requirements and identifying relevant personnel. Furthermore, a number of regulatory documents need to be completed. Once committed to the project, servers are purchased.

2. Developing the file extracts

While waiting on server acquisition, begin development of extracts that pull anesthesia data from the source system.

3. Setting up the MPOG Infrastructure

Once the servers are ready, the MPOG components are installed and tested. This stage culminates in the upload of data to the central repository.

4. Ongoing Maintenance

With the setup complete, monthly tasks and troubleshooting protocols are reviewed.

Key Concepts

Going forward, all new MPOG implementations will use the Import Manager framework. The purpose of this framework is to enable hospitals to easily extract their anesthesia data for MPOG's research and quality improvement initiatives. From a single hospital database to a complex multi-instance conglomerate, the Import Manager is designed to handle a wide range of possible configurations. Additionally, this framework attempts to automate as many processes as possible, such as handling updates to already extracted data and applying changes in variable mapping.

For these reasons, the Import Manager is somewhat complex and contains numerous components. In order to better acquaint you with this framework, the high-level concepts are described in the sections below.

Local vs Central Database

There is more than one copy of an institution's MPOG data. There is a local copy that is stored within an institution's network as well as being uploaded to MPOG's central repository. The local copy contains only that institution's data while the central repository contains the data of all contributing institutions.

One important difference between these data sets is that the local copy contains PHI while the central one does not. The upload process leaves out patient identifiers such as name, date of birth, and SSN. The primary reason PHI is kept locally is for scrubbing these identifiers out of text fields found throughout the anesthesia record.

File Pipeline

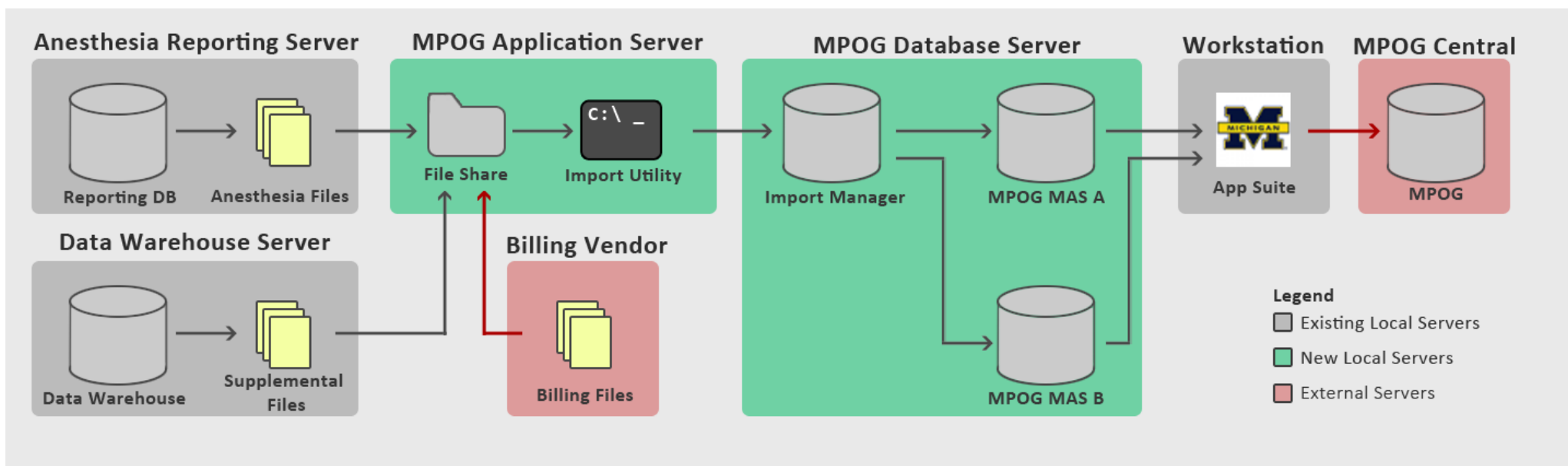
MPOG collects data from source systems (e.g. data warehouses, third-party billing vendor, lab interface, etc.) through the use of comma delimited text files. These files flow through a number of different systems and are transformed several times prior to ultimately being uploaded to MPOG's central repository. This process is generalized into the following five steps:

1. *Extract* – Files are generated by extracting data from its source system. These files are then placed into a file share accessible by the MPOG import utility.
2. *Import* – Files are removed from the file share and inserted directly into the Import Manager database.
3. *Consume* – File stored within the Import Manager are parsed into tabular data and inserted into staging tables. Additionally, metadata regarding variable usage is generated for later mapping.

4. *Handoff* – The tabular data is inserted into the designated MPOG MAS database.
5. *Upload* – Using the transfer utility within the MPOG application suite, surgical cases are uploaded to the central repository.

Please see the File Pipeline diagram below for a graphical representation of this process.

File Pipeline Diagram



Note that this diagram is only one example of many possible configurations. A number of factors may affect your institution's pipeline including:

- The number of instances of source anesthesia system
- The number of billing vendors
- The number of MPOG_MAS databases
- The number of user workstations with the application suite installed

Instances

An “instance” within the Import Manager framework broadly refers to a source anesthesia system. The framework supports multiple instances, whether they are multiples of the same type or a combination of current and legacy systems.

The main impact of having multiple instances on the file pipeline is that each instance from must have a separate directory. These directories are then assigned to an instance during setup.

Note: Given the possibility of identifier collision, data from different instances cannot be combined. As such, cases from different instances cannot be assigned to the same MPOG MAS database.

Organizations

Aside from instances, “organizations” are the primary method by which the framework splits apart cases. For multiple hospitals sharing the same instance, it is often desirable to separate their cases into different databases. This allows each institution to validate their own cases and prevents users from sister institutions examining their data.

Each surgical case is assigned an organization by the extract. All organizations detected by the framework can be assigned a destination MPOG MAS database. Organizations can be assigned to the same database, different databases, or no database at all. An organization by default is assigned to no database and therefore its cases will not leave the Import Manager.

Note: Each MPOG MAS database has its own validation requirements regarding monthly diagnostic attestation and cases to validate. Given organizations tend to be quite granular, it is not advisable to create one MPOG MAS per organization. Doing so would create a huge volume of work for the clinical reviewer.

Variable Mapping

Data elements are commonly stored in anesthesia systems with an assigned variable. These variables represent a wide variety of things such as blood pressures, medications, notes, staff roles, units of measure, and so on. Different anesthesia systems use different variables and even the same anesthesia system can have variables configured differently between hospitals.

In order to provide a standardized lexicon for all these different elements, MPOG provides a data dictionary called MPOG concepts. Many of the variables used within your anesthesia system will be mapped to a MPOG concept. This mapping helps the Import Manager framework to decide where a certain element should go (e.g. intraoperative note vs monitor recording). Furthermore, it helps queries to be written for the central repository without knowing each institution’s specific variables.

Variable mapping is performed primarily by the clinical reviewer using the provided mapping utility. For more information on how to do mapping, refer to the Variable Mapping Guide and your MPOG clinical support.

Validation

In order to ensure the data being extracted is correct, there are two main methods of validation performed by the clinical reviewer. The clinical reviewer will be responsible for completing this validation each month.

Diagnostics is an application that displays macro-level views of data quality. It shows fill rates and mapping rates over time for many different classes of data. These high-level tests are useful for detecting any systemic issues with the extract and checking if there are any missing elements.

Case Validation is another application which allows the user to check individual cases and compare them to their source system. The application provides a checklist of questions to quickly guide through checking individual fields such as medications given and blood pressures recorded.

Stage 1: Laying the Groundwork

The regulatory and legal requirements are a critical step to MPOG implementation. Some of the approval processes can be challenging and can weeks to months to complete.

1-1. Apply for Membership

Implementation of MPOG requires a high level of technical expertise to create the technical infrastructure needed to submit data to MPOG. An institution must ensure that the department chair or head of practice is aware of the financial and FTE requirements of MPOG implementation and supports joining MPOG. Please see [FTE Requirements and Cost](#) for more details.

Once you have identified the personnel for this project, please fill out the relevant application below and return it to the MPOG Program Manager:

- [MPOG Application: BCBSM Funded](#)
- [MPOG Application: Non-BCBSM Funded](#)

1-2. Complete a Business Associate Agreement (BAA)

A Business Associate Agreement (BAA) is a legal document that must be put in place between the Regents of the University of Michigan and your institution.

This is normally done through a grants/contracts office at each institution but may vary depending on the site. This agreement is put in place because during the technical implementation and training phases. MPOG developers and clinical staff may be exposed to protected health information (PHI) from your site. This document is put in place to allow MPOG staff to properly train staff.

You may start with [University of Michigan's BAA template](#), but many sites wish to use their own.

This step MUST be completed prior to installing the MPOG infrastructure (Stage 3).

1-3. Complete a Data Use Agreement

Data use agreements (DUA) are drafted between the University of Michigan's and your institutional attorneys. This is normally done through a grants/contracts office at each institution but may vary depending on the site. Understand that these agreements are at the institutional level (not departmental or individual level) and will therefore require signatures from institutional representatives.

MPOG DUA templates are included below. Please select the relevant DUA based upon your location.

- [Master DUA - United States](#)
- [Master DUA - Europe](#)

The DUA must be in place before data submission to MPOG Central Repository.

1-4. Submit Performance IRB Application for Approval

Prior to beginning a research project using your institution's data, you will need to apply for Institutional Review Board (IRB) approval. Refer to our [IRB article](#) for more information. Download the [IRB Application](#) to get started. Be aware that IRB approval can take weeks to months to obtain, depending on your institutional IRB process.

Note that if your institution does not have an onsite IRB, FDA regulations permit an institution without an IRB to arrange for an 'outside' IRB to be responsible for initial and continuing review of studies conducted at the non-IRB institution. The University of Michigan can accept oversight for non-exempt research.

1-5. Review MPOG Bylaws and Terms of Membership Participation

You will need to review the [MPOG Bylaws](#) and agree to the terms of membership. Sign the agreement form and return to the MPOG Program Manager.

1-6. Review Data Requirements

Prior to submission, MPOG requires that a contributing institution must have at least 10,000 anesthetic cases. This generally corresponds to at least six to twelve months' worth of cases. Furthermore, all data to be submitted must satisfy the minimum data requirements, which can be found at www.mpogresearch.org/minimum-data-requirement.

Because of this requirement, institutions should not start on MPOG implementation until at least a few months after go-live. We recommend waiting until the system is stable, so the changes to the build are minimal. The implementation impact for an MPOG interface is considerable. We estimate that it will take at least six months to implement the project. Specific timelines might vary depending on the availability of your staff.

Institutions seeking to participate with ASPIRE have an additional requirement of submitting at least twelve months' worth of data, regardless of case volume.

1-7. Acquire Servers

You need a local database server to store MPOG data and submit to the national MPOG server. Additionally, you will need a modest application server to host some applications and services related to

automation. Work with your IT department and your MPOG Database Administrator to purchase and configure a server that meets the requirements outlined below.

Database Server

- Physical or Virtual Machine
- CPU: 4+ cores (faster cores are recommended over more cores)
- RAM: 8+ GB minimum (32+ GB recommended)
- Storage: 500 GB is the recommended starting point (each surgical case is ~2 MB)
- Microsoft Windows Server 2012 R2 or newer
- Microsoft SQL Server 2012 or newer
 - Server Collation: SQL_Latin1_General_CP1_CI_AS (non-Case Sensitive)
- Microsoft .Net Framework 4.5 or later

Application Server

- Physical or Virtual Machine
- CPU: 2 cores minimum
- RAM: 4-8 GB
- Storage: 50 GB
- Microsoft Windows Server 2012 R2 or newer
- Microsoft .Net Framework 4.5 or later

1-8. Find Sources of Supplemental Data

Not all data elements required by MPOG are currently implemented in the Chronicles extract. These supplemental data sets are then pulled from other systems.

Professional Fee Billing Codes

Anesthesia professional fee billing codes are sometimes found in Clarity. However, they are often handled by a third-party billing vendor. If your institution is comprised of multiple hospitals or anesthesia groups, you may have more than one vendor.

It's important to determine the source of these billing codes quickly. If data is pulled from a vendor, they will need time to develop an extract to MPOG's file specifications.

Hospital Discharge Billing Codes and In-Hospital Dates of Death

Hospital discharge billing codes and in-hospital dates of death are commonly found within Clarity. If so, starting queries will be provided by your MPOG Technical Support. Again, it's important to identify the source of this data quickly to determine if any modifications to the provided queries are necessary.

1-9. Request an Application Account

In order for the Import Utility to be run automatically, it will need an application account for it to run under. Work with your Active Directory support team to create this account. Given this can take some time to process this request, it is recommended doing this early on in the project.

Stage 2: Develop the File Extracts

Before data can be imported into the MPOG framework, first it must be dumped into specifically formatted comma-delimited files. Unfortunately, given the large number of healthcare documentation systems MPOG cannot provide or develop an extract for every hospital. As such, an organization wishing to participate with MPOG must develop their own extract to generate the data files. This section provides a high level overview of this development.

2-1. Review File Specification

In order for Import Manager to correctly process data files, the files must be structured correctly. These rules include how the files must be named, which columns are in each file, and how each column must be formatted. The details on how to structure the data files can be found in a separate document, the Import Manager File Specification.

Main Anesthesia Files

The main anesthesia files primarily contain documentation that was recorded by an anesthesiologist or that was relevant to the anesthesiologist's workflow. Whenever main anesthesia files are mentioned, the following modules are included:

- Patients
- Cases
- Labs
- StaffTracking
- PeriopObservations
- PeriopObservationDetails
- PeriopAdministrations

Supplemental Files

Supplement files are called as such because they contain data typically not found within a dedicated anesthesia system. It is not uncommon to obtain this data from a data warehouse or third party biller. When supplemental files are mentioned, the following modules are included:

- HospitalMortality
- Procedures
- ProcedureModifiers
- Diagnoses
- Payers

2-2. Develop Extracts

There are many ways to go about extracting data into files. Generally, this requires two problems to be solved. First is querying the data from various sources and transforming to fit the MPOG specification. This is most often accomplished through the use of a database query language such as SQL. Second is writing these data sets to a file in an automated fashion. How this is done is much more varied, as there are many tools and programming languages that can read from a database and write to a file. Common examples are Python, SSIS, and some variants of SQL such as T-SQL.

Which approach to use in solving these two problems is dependent on a number of factors such as database type, developer skillset, and institutional policy. Your MPOG Technical Support can offer advice regarding your organization's specific needs.

Once development is ready to begin, it is recommended to start with the Cases and Patients modules, as it is not possible to test the other modules without having these base modules in place. From there, the rest of the main anesthesia files should be developed. The supplemental files require the least amount of validation and mapping, so it is recommended to leave those for last.

2-3. Setup File Transfer

Typically, when files are generated they are not placed directly into the file share accessible by the MPOG application server. Instead, they are written to a server local to where the extract is running. This server may even be on a different network. Therefore, it is common to setup a process which moves these files from where they were initially placed to where they can be processed by the MPOG application server.

Most organizations utilize FTP or SFTP for this transfer. Consult your local IT team as this is a common task and a workflow for creating this process may already exist.

2-4. Automate Extract Execution

During the early stages of the setup phase the extracts will be run manually. However, the extract will need to be automated to pull in a year's worth of cases as well as to accommodate the daily maintenance schedule of pulling in new and updated data.

As with the extracts themselves, there are a large number of ways to automate them. Two common ways are through either SQL Server's job agent or Windows' Scheduled Tasks.

Stage 3: Setting up the MPOG Infrastructure

Once the MPOG Database Server and MPOG Application Server are ready, you can begin the setup of the MPOG infrastructure (refer to Stage 1: Laying the Groundwork). These steps will primarily be carried out by the Local Technical Support, unless otherwise specified.

3-1. Request Installation Package

Request an installation package from your MPOG Technical Support to get started.

3-2. Enable CLR Execution

Run the following SQL on the MPOG SQL server to enable the execution of "SAFE" CLR assemblies. These assemblies will be installed later and are chiefly used for the efficient processing of text files.

```
--Note: Running this script requires sysadmin privileges.  
sp_configure 'clr enabled', 1;  
GO  
RECONFIGURE;  
GO
```

3-3. Create Blank Databases

Create a blank database called MPOG_Import_Manager.

In addition, create a blank database for each institution you plan to extract (see the “Organizations” section). Each of these databases should be prefixed with “MPOG_MAS_” and suffixed with a short name (< 10 characters) describing the institution. Some examples are MPOG_MAS_AnnArbor or MPOG_MAS_Detroit.

For the above databases, we recommend the following configuration:

- Arrange the database files according to your institution’s policies
- Set recovery mode to “Simple”
- Weekly full backups to another server. Nightly differentials are also recommended.

3-4. Run Installation Scripts

Ask your MPOG technical contact for the most up-to-date installation scripts. These scripts will be grouped by database, MPOG_Import_Manager and MPOG_MAS. Each script will be prefixed with a number

Run the scripts for the MPOG_Import_Manager database in sequential order. If an error is returned, stop and ask your MPOG technical contact for assistance.

Do the same for the MPOG_MAS scripts, except these you will against each MPOG_MAS_ database you created.

3-5. Insert an Institution Definition Row

Each MPOG_MAS database has its own unique identifier. Tell your MPOG technical contact your institution names and they will provide you with the appropriate INSERT statements.

```
--Note: This is an example only.  
USE MPOG_MAS_Fake  
GO  
INSERT INTO dbo.MPOG_Institutions  
VALUES (123.456, 'Fake Hospital Name', FHN', '2017-01-01')  
GO
```

3-6. Setup File Import Utility

The installation package provided by MPOG will contain the Import Utility archive. Make sure both the MpogFlatFileImporter .exe and .config files are extracted and place them in a directory ("C:\Program Files\MPOG\" is recommended).

In the config file, update the server address in the connections strings section (denoted by "SERVERADDRESSHERE") to the address of the database server.

Also, note that the dateFormat setting should match the "Target Date" first column in the multi-date files. This setting is not used for dates in other columns in multi-date files and note at all for single date files. You can find additional details on the string formats at [https://msdn.microsoft.com/en-us/library/8kb3ddd4\(v=vs.110\).aspx](https://msdn.microsoft.com/en-us/library/8kb3ddd4(v=vs.110).aspx).

3-7. Create Extract Folder(s)

The Import Utility will read files from one or more folders, insert them into the MPOG_Import_Manager database, then delete the files from the folder. Each folder will correspond to a single extract instance (see the "Instances" section).

Typically, these directories are located on the application server on a non-system drive (e.g. "D:\MPOG_Extracts\"). However, they can be created elsewhere. The main requirement is that the Windows account(s) running the Import Utility have read/write access to that location.

3-8. Configure Extract Instance(s)

Here you will configure the file directory for each extract instance. Each instance will have its own ID, name, and file path. Most installations only require a single instance. For each instance, modify the SQL below and run it against MPOG_Import_Manager.

```
INSERT INTO Config.Instance_File_Paths
(
    Instance_ID,
    Instance_Name,
    File_Path
)
VALUES
(
    0,
    '<INSTANCE NAME>',
    '<FULL FILE PATH HERE>'
)
```

Instance_ID – This integer is the main way the instance is referenced throughout the MPOG_Import_Manager database. This ID is manufactured and doesn't correspond to anything outside the database.

Instance_Name – This name is used to differentiate the instances to developers.

File_Path – This path is the location of the folder that will contain this instances files once extracted. Note this location is in relation to the Import Utility, not the database server.

3-9. Configure User Database Access

The application suite relies on permissions set on the database to determine which applications the user can use. This access is controlled through pre-defined database roles, which are detailed in the [MPOG Database Roles](#) document. These roles can be granted to either individual logins or to active directory groups. The following access settings are recommended:

- Clinical Reviewer & Clinical Champion
 - MPOG_VariableMapper
 - MPOG_Exporter
 - MPOG_Researcher
 - MPOG_CaseValidator
 - MPOG_ContentDownloader
- Local Technical Support
 - db_owner on all MPOG databases

- MPOG Technical Support (optional)
 - db_owner on all MPOG databases
- Local MPOG Database Administrator
 - N/A (assumed to have sysadmin server permission)

3-10. Install Application Suite on User Workstation(s)

The application suite contains various utilities for use with the MPOG databases. It is installed locally to the Windows user profile and usually can be installed directly by the user. The [setup executable](#) will automatically install the latest available version.

However, some institution's desktop environment prevents this type of installation. If this is the case, work with your workstation IT and MPOG Technical Support to create a custom application package.

3-11. Extract Files

For the first extract, pull only a single day's worth of data. This will be sufficient to test the rest of the file pipeline, making sure there are no issues prior to pulling more files. Also, pulling smaller amounts of data means a smaller cycle time for applying fixes and re-extracting. The typical extract progression is as follows:

1. One Day
2. One Week
3. One Month
4. All data

3-12. Move Files

Once the files have been extracted, move them to the extract folder created in step 3-7.

During the first few tests of the pipeline, create a backup copy of the files to a subfolder. In the event something goes wrong and the files are accidentally lost or corrupted, you won't have to wait to re-extract the files.

3-13. Run Import Utility

Run the Import Utility, either by double-clicking on the EXE or by running it in the command line with no arguments. You will see each filename listed as it is processed by the utility. The files will also be deleted from the folder once it is successfully imported.

You can check if the files were imported into the database by running the following queries.


```
--These tables also have a "Data" column that contains the text of the file
--It is omitted for performance
SELECT
    Instance_ID,
    Imported_File_Name,
    Imported_Time
FROM
    Import.Files

SELECT
    Instance_ID,
    Imported_File_Name,
    Inserted_Time,
    Completed_Time
FROM
    Import.PartialFiles
```

If the Import Utility exits unexpectedly or files remain in the extract folder once it completes, contact your MPOG technical support for assistance.

3-14. Consume Files

The file data is now ready for parsing. Run the following script to parse the files into their respective staging tables.

```
EXEC MPOG_Import_Manager.Core.ConsumeFiles
```

This will take several minutes to a few hours, depending on the amount of data being consumed. You can check on its progress with the following query.

```
SELECT
    *
FROM
    Core.Log_Consume
ORDER BY
    Processing_Start DESC
```

You can also check the staging tables in MPOG_Import_Manager to make sure data is being parsed correctly.

```

SELECT COUNT(*) FROM Staging.Patients_V1
SELECT COUNT(*) FROM Staging.Cases_V1
SELECT COUNT(*) FROM Staging.PeriopAdministrations_V1
SELECT COUNT(*) FROM Staging.PeriopObservations_V1
SELECT COUNT(*) FROM Staging.PeriopObservationDetails_V1
SELECT COUNT(*) FROM Staging.StaffTracking_V1
SELECT COUNT(*) FROM Staging.Labs_V1

```

3-15. Configure Organization's Destination Database

Each organization detected by the infrastructure can be split into the different MPOG_MAS (see the "Organizations" section for more information). Now that at least one Cases file has been imported and consumed, organization metadata is available. Check the Config.Organizations table to see which organizations have been imported.

```

SELECT * FROM Config.Organizations

```

By default, the Destination_Database and MPOG_Institution_ID columns are left NULL. This means that no organization's data leaves the MPOG_Import_Manager without being explicitly configured to.

Organizations can be configured to export to a specific MPOG_MAS. You can modify the following SQL to update the organizations based upon their name.

```

UPDATE
  MPOG_Import_Manager.Config.Organizations
SET
  Destination_Database = 'MPOG_MAS_Fake',
  MPOG_Institution_ID = 123.456
WHERE
  Organization_Name = '<CONDITION>'

```

Note that not all organizations need to be configured. In fact, it is often desirable as non-participating organization's data can be filtered out from being exported to a MPOG_MAS.

3-16. Configure user access for mappings

Before the Clinical Reviewer can use the variable mapping utility, they will need to be granted the appropriate permissions. In addition to the permissions set in step 3-9, they will also need to be granted access to specific organizations.

Variable mapping can be partitioned by organization, with each user only able to modify mappings relevant to their organization. However, it is most common for a single clinical reviewer to be in charge of all mappings. In this case, the following script will grant them access to all organizations.

```

DECLARE @username VARCHAR(100)

--Modify this to the user's name (typically includes the domain)
SET @username = '<DATABASE_USERNAME>'

INSERT      INTO Config.User_Access
      (
        Organization_ID,
        User_Principle
      )
SELECT DISTINCT
      Organization_ID,
      @username
FROM
      Config.Organizations o
LEFT JOIN config.User_Access ua
ON ua.Organization_ID = o.Organization_ID AND
ua.User_Principle = @username

WHERE
      ua.User_Principle IS NULL

```

3-17. Map variables to MPOG Concepts

The clinical reviewer can now map variables detected in the files to MPOG Concepts. They can refer to the Variable Mapping guide for more details.

MPOG Configuration

Variable Mapping | Administration Totalling | Observation Entry

Mapping Type: **Observation Type**

Organization: **Epic**

Display Mode: **Mapped Variables Only**

Search Filter:

Direction: **Normal**

Options: **Auto Search On** **Auto Select On**

Search Filter:

ID	Name	Times Used	Mapped As
LEV-890000	Anesthesia Start	221	AACD Anesthesia Start Date/Time
CT-70	Anesthesia Start	214	AACD Anesthesia Start Date/Time
LEV-890001	Anesthesia End	197	AACD Anesthesia End Date/Time
CT-100	Anesthesia Stop	191	AACD Anesthesia End Date/Time
FLO-2201	Resp	174	Respiratory - Other
FLO-2648	O2	159	Flows Oxygen (L/Min)
FLO-2186	ETCO2	158	Respiratory Rate Actual from ETCO2
FLO-911004517	FIO2	158	Ventilator FIO2 % Set
FLO-2649	N2O	153	Flows Nitrous Oxide (L/min)
HLX-EPIC#0251	anesthesia plan	149	Assessment and Plan - Anesthesia
FLO-2650	Sevoflurane	148	Sevoflurane actual consumption (ml)
FLO-2651	Isoflurane	136	Isoflurane actual consumption (ml)
FLO-3769	ANTP_Air	123	Flows Air (L/min)
FLO-3770	ANTP Carbon Dioxide	118	Inspired CO2 %
FLO-2655	Tidal Volume	117	Tidal Volume actual
FLO-2198	Inv BP	114	BP Mean Invasive Unspecified Site
HLX-EPIC#1004C	ASA status	91	Assessment and Plan - ASA Physical
FLO-2199	NInv BP	78	BP Mean Non-invasive
FLO-2200	Pulse	58	EKG Pulse Rate
FLO-2203	Temp	46	Temperature - Skin

Name	ID
EKG Pulse Rate	3005
Isoflurane actual consumption (ml)	3006
Desflurane actual consumption (ml)	3007
Sevoflurane actual consumption (ml)	3008
BIS - Suppression Ratio	3009
SpO2 Pulse Rate	3010
BP Sys Invasive Unspecified Site 1	3011
BP Dias Invasive Unspecified Site 1	3012
BP Mean Invasive Unspecified Site 1	3013
BIS - Total Power	3014
BP Sys Non-invasive	3015
Pulse Pressure Variation	3016
Regional Oxygen Saturation	3017
Regional Oxygen Saturation Baseline	3018
Stroke Volume Variation	3019
BP Dias Non-invasive	3020
Non-invasive Total Hemoglobin Concentration	3021
Stroke Volume	3022
Stroke Volume Index	3023
Stroke Volume Variance Index	3024
BP Mean Non-Invasive	3025

Map Unmap Exclude Examine

Screenshot of the Data Diagnostics utility

Import Existing Mappings (Optional)

If you have existing variable mappings from a legacy process, contact your MPOG technical support for assistance.

3-18. Run Handoff to MPOG_MAS databases

Data is now ready to be exported from the MPOG_Import_Manager to the appropriate MPOG_MAS. This process is called “Handoff” and can be initiated by running the following two stored procedures.

```
--Determines which modules and target dates to run
EXEC MPOG_Import_Manager.Handoff.ScheduleItemsOntoQueue

--Processes items on the queue
EXEC MPOG_Import_Manager.Handoff.ProcessQueue
```

Again, this process will take anywhere from a few minutes to several days, depending on the amount of data needing processing. While this process runs, the following query can be used to see which items are currently on the queue.

```
SELECT * FROM MPOG_Import_Manager.Core.Handoff_Queue ORDER BY Priority
```

In addition, the following query displays the handoff’s progress, which module is currently processing, and if any errors have been encountered.

```
SELECT * FROM MPOG_Import_Manager.Core.Log_Handoff ORDER BY Handoff_Start DESC
```

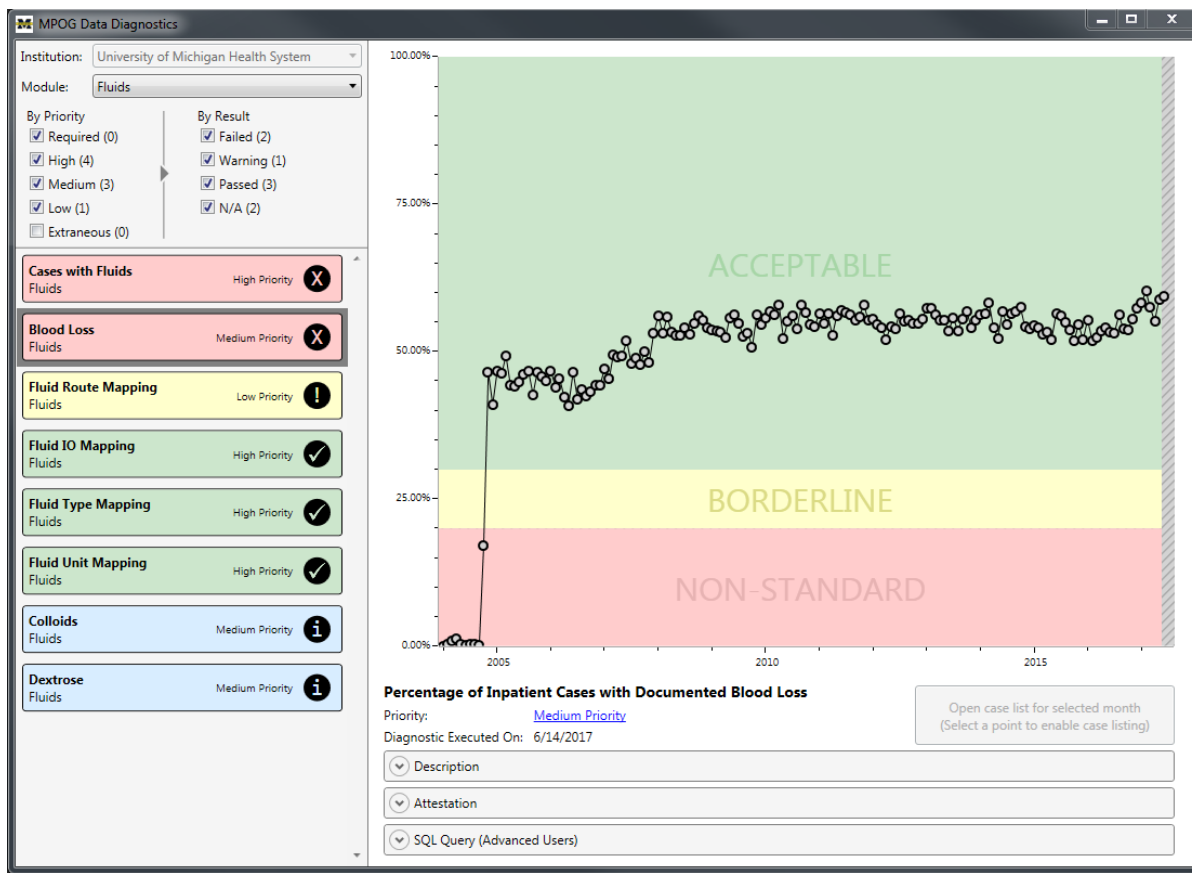
In the early stages of the project it is recommended to also verify data is showing up in the MPOG_MAS database(s).

```
USE MPOG_MAS_MyHospital --change this to your MPOG_MAS name
GO

SELECT COUNT(*) FROM AIMS_Patients
SELECT COUNT(*) FROM AIMS_IntraopCaseInfo
```

3-19. Run Diagnostics

Once handoff is complete, MPOG’s built-in diagnostics can be run to assess data quality. These diagnostics run various checks and cache the results. The clinical reviewer can then check these results in Data Diagnostics utility in the application suite.



Screenshot of the Data Diagnostics utility

You can run diagnostics with the following script.

```
--Run the following for each MPOG_MAS database
EXEC Eval.Diagnostics_RunAllDiagnostics
```

Running diagnostics typically takes 10-30 minutes, but can take longer if you have a significantly higher case volume.

3-20. Validate Data and Update Mappings

The clinical reviewer can now begin to examine and validate cases, using the Case Viewer, Data Diagnostics, and Case-by-Case Validation utilities. They can refer to the Validation Guide for more details.

As the clinical reviewer finds issues through their validation process, they may find variables that were missed in the first pass or variables that were mapped to the wrong concept. They can update these mappings using the variable mapping utility and the infrastructure will handle the necessary updates.

3-21. Give Application Account Necessary Permissions

The account the Import Utility will run as will need a login on the MPOG database server and a corresponding user account on the Import Manager database. Assign it the MPOG_ImportUtility database role to ensure it has the necessary permissions.

3-22. Setup Pipeline Automation

Now that the pipeline has successfully been tested, you can automate many of the processing steps. After these components are configured, files that appear in the extract folder will automatically be processed with little to no involvement from the database administrator.

SQL Agent Jobs

The database server's SQL Agent is used to schedule and run the following processes:

- Consuming files
- Scheduling the handoff queue
- Processing the handoff queue
- Running diagnostics

Work with your MPOG technical support to create these jobs and appropriate schedules for each.

Note that modifying SQL Agent will require sysadmin access on the database server.

Scheduled Tasks

The Import Utility is automated using Windows Task Scheduler. Create a task called "MPOG – Run File Import Utility" that starts the utility. Schedule it to run every five minutes. Configure it to run using the credentials created in step 19.

3-23. Expand Extract Scope

Repeat steps 3-11 to 3-20, increasing the amount of data extracted as detailed in step 3-11 until a pull of all historical data has been completed.

3-24. Setup Regular Extracts

Now that the extracts have been tested, they can be automated. The setup varies based upon the originating system, so refer to the appropriate section below.

If pulling data from a third-party vendor, ongoing extracts are typically scheduled on a monthly basis. Work with your billing vendor representative to schedule these extracts and to setup an automated SFTP process for transferring the files.

3-25. Meet with MPOG for Review

Once the minimum requirement of data has been extracted and validated, meet with MPOG to perform an assessment of data quality. This meeting will include a MPOG clinical coordinator and your institution's clinical champion(s). These parties will identify any issues for uploading to the central repository.

If any outstanding issues are determined to be high priority, they will need to be resolved prior to continuing.

3-26. Upload to Test Repository for MPOG Director Review

The final sign-off of data quality is performed by MPOG leadership. To do this data must be uploaded to our test repository. Run the PHI scrubbing utility to prep cases for upload and run the transfer utility with the Database Selection option set to "developer".

Refer to the App Suite Guide for additional information.

3-27. Complete Outstanding Validation

If at this point there is any outstanding validation, the Clinical Reviewer will need to complete it prior to upload as a final check step.

3-28. Upload to MPOG Central

The Clinical Reviewer is now ready to upload to the production central repository. Run the PHI scrubbing utility to prep cases for upload. Run the transfer utility with the default settings to complete the upload.

Refer to the App Suite Guide for additional information.

Stage 4: Ongoing Maintenance

Once you leave the setup stage, the majority of the work has been completed and the processes automated. However, there are some ongoing, manual maintenance tasks to be aware of.

Clinical Reviewer Monthly Validation

The bulk of the ongoing effort is taken by the clinical reviewer. He or she is responsible for the continued validation of each month's data as well as upload. If your institution is a member of ASPIRE, the clinical reviewer will also be responsible for reviewing cases that may have failed ASPIRE measures.

Schema Updates

Roughly twice a year, MPOG will release updates to its database schema. Once available, your MPOG technical support will schedule a brief teleconference with your local MPOG DBA to install the upgrade. These upgrades typically take 10-20 minutes to complete.

Server Maintenance and Light Troubleshooting

Your institutional IT is responsible for normal server maintenance, such as applying Windows updates and taking database backups.

Additionally, your support staff will handle issues that may arise. Refer to the MPOG Troubleshooting Guide for details for the appropriate contact for common issues.

Appendix A: Setup Phase Timeline

An Excel spreadsheet of this timeline is also available.

Step	Description	Section	Dependencies	Performed By	Estimated Time
1	Complete membership application	1-1	None	Project Manager	4 - 6 weeks
2	Complete BAA	1-2	1	Project Manager	4 - 6 weeks
3	Complete DUA and IRB	1-3, 1-4	1	Project Manager	4 - 6 weeks
4	Acquire servers to host MPOG backend	1-7	1	Server Acquisition	4 - 6 weeks
5	Contact billing vendors	1-8	1	Project Manager	4 weeks
6	Contact reporting team for mortality and hospital billing	1-8	1	Project Manager	4 weeks
7	Request application account	1-10	1	Project Manager	4 weeks
8	Develop file extract	2-1 to 2-2	1	Local Technical Support	6 - 8 weeks
9	Setup file transfer process	2-3	1	Local Technical Support	2 - 4 weeks
10	Install MPOG framework	3-1 to 3-8	2	Local Technical Support	1 week
11	Install MPOG suite on user workstation(s)	3-14 to 3-16	10	Local Workstation IT	1 week
12	Pipeline test – one day		8, 10	Local Technical Support	1 week
	a Extract main anesthesia files	3-11			
	b Extract pro fee billing (optional)	3-11			
	c Extract hospital billing (optional)	3-11			
	d Extract in-hospital mortality (optional)	3-11			
	e Move files	3-12			
	f Import	3-13			
	g Consume	3-14			
	h Handoff	3-15, 3-18			
13	Pipeline test – one week		12	Local Technical Support	1 week
	a Extract main anesthesia files	3-11			
	b Extract pro fee billing	3-11			
	c Extract hospital billing	3-11			
	d Extract in-hospital mortality	3-11			
	e Move files	3-12			
	f Import	3-13			
	g Consume	3-14			
	h Handoff	3-15, 3-18			
14	Automate handoff and diagnostic processes	3-22	13	Local Technical Support	1 day
15	Configure user access for variable mapping	3-16	13	Local Technical Support	1 day
16	Variable mapping - first round	3-17	15	Clinical Reviewer	2 weeks
17	Check diagnostics	3-20	16	Clinical Reviewer	1 week
18	Pipeline test – one month		17	Local Technical Support	1 week
	a Extract main anesthesia files	3-11			
	b Extract pro fee billing	3-11			
	c Extract hospital billing	3-11			
	d Extract in-hospital mortality	3-11			
	e Move files	3-12			
	f Import	3-13			
	g Consume	3-14			
19	Variable mapping - additional round	3-17	18	Clinical Reviewer	1 week
20	Check case viewer and diagnostics	3-20	19	Clinical Reviewer	1 week
21	Perform case validation	3-20	20	Clinical Reviewer	2 weeks
22	Automate import and consume processes	3-22	18	Local Technical Support	1 day
23	Repeat steps 18-21 until full year is completed	(various)	21	(various)	6 weeks
24	Automate file extract	2-4, 3-24	23	Local Technical Support	1 week
25	Review of data with MPOG and clinical leadership	3-25	23	Clinical Reviewer	1 week
26	Resolve any issues from review meeting	(various)	25	(various)	3 weeks
27	Upload data to test repository for MPOG director review	3-26	26	Clinical Reviewer	1 day
28	Resolve any issues from MPOG director review	(various)	27	(various)	3 weeks
29	Complete outstanding validation	3-27	28	Clinical Reviewer	1 week
30	Upload data to MPOG central repository	3-28	29	Clinical Reviewer	1 day

Appendix B: Troubleshooting Guide

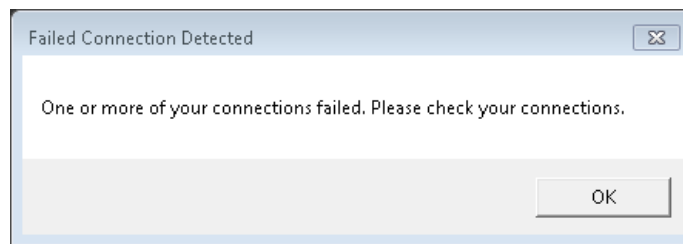
This document lists common problems encountered with the MPOG infrastructure. Each problem includes the typical symptoms and steps that can be taken to resolve them.

If an issue cannot be resolved, the final step will list the most appropriate support contact to reach out to. If you are having an issue not listed below, please contact your MPOG Technical Support directly.

MPOG Suite General - Startup Connection Issue

Symptoms

The MPOG application suite has issues at startup connecting to the database, presenting the following dialog box:



Error screen associated with this error

Steps to Try (End User)

1. Click OK. This will bring you to the profile management screen.
2. Click "Edit Selected" to edit your current connection profile.
3. Verify the Server and Database fields are correct. You may need to check with your IT contact for the latest information.
 - a. Server: _____
 - b. Database: _____
 - c. Note that Config and Research connections are unused and should be left blank.
4. Click OK. If no warning screen appears, the connection worked.
5. If a warning screen detailing the status of the different connection types appears, send a screenshot to your local technical contact.

Steps to Try (Technical Support)

1. If not already provided, ask the user for the error message returned by the suite. This will help identify the issue more quickly, such as:
 - a. User has no login
 - b. User has lacks permissions
 - c. Timeout error
2. Verify the user's database connection information is correct.

3. Verify the database server is available.
4. Verify the relevant MPOG_MAS database is available.
5. Verify the user has a login on the server.
6. Verify the user has the correct permissions on the database, at a minimum having the MPOG_Researcher role.
 - a. Additional details regarding the MPOG database roles can be found on the MPOG website
(https://www.mpogresearch.org/sites/default/files/downloads_technical/Guides/MPOG%20Database%20Roles.docx).
7. If still not resolved, reach out to MPOG support.

MPOG Suite - Crash Error

Symptoms

The MPOG suite stops working and presents a crash error report screen.

Steps to Try

1. Fill out the fields in the error report, detailing the steps leading up to the crash if possible.
2. Click “Send Error Report” and MPOG support staff will get back you regarding the error.
3. If for some reason the report fails to send, email your MPOG support contact directly.

Upload Utility - Uploading to Central Error

Symptoms

While uploading cases to the central MPOG repository, the upload utility presents an error stating “A serious error was encountered and the upload was aborted.”

Steps to Try

1. Check if internet access is available by opening a website (<http://www.med.umich.edu/>)
 - a. If not, find internet access either by connecting to a wireless network or by contacting your IT department.
2. Wait a few minutes, restart the suite and try the upload again. Network issues can sometimes interrupt the upload.
3. If the issue persists, reach out to your MPOG support contact.

Diagnostics – Results are Out of Date or Have Not Been Run

Symptoms

The diagnostics have not been run recently and more up-to-date results are needed. (Note the diagnostic last run time is listed underneath the diagnostic chart.

OR

All diagnostics have a gray background with a “NO DATA AVAILABLE” message.

Steps to Try (End User)

1. Running diagnostics is handled on the backend and cannot be started by a normal user. Reach out to your local technical contact to have them run the process.

Steps to Try (Technical Support)

1. If diagnostics results need to be refreshed immediately, they can be executed by running the following SQL on each of your MPOG_MAS databases:
 - a. EXEC Eval.Diagnostics_RunAllDiagnostics
2. Diagnostics should be run every night automatically. Check the MPOG diagnostic job in the SQL Agent for the following:
 - a. Verify that SQL Agent is running.
 - b. View the job history to determine if the job is failing or not being run at all.
 - c. Verify that both the job and its schedule are enabled.
 - d. Verify the job contains the SQL mentioned above.
3. If still not resolved, reach out to MPOG support.

Diagnostics \ Case Validation – Missing or Incorrect Data

Symptoms

Data expected from the anesthesia record is not appearing in data diagnostics or case validation. The following are some examples of this issue:

- Diagnostics show lower fill rates than expected
- Case Validation lists “NOT FOUND” for a given question
- A common or expected data element is missing from the Case Viewer
- Case Validation question is incorrect i.e. answered “No”
- Data displayed within the Case Viewer is incorrect

Steps to Try (End User)

1. Verify data is in original anesthesia record. Sometimes data elements are genuinely missing from the original documentation or were documented incorrectly.
2. Check variable mapping, if applicable. Many MPOG applications rely on accurate variable mapping to identify different types of data. It’s possible that one or more relevant variables have not been mapped or was mapped to the wrong concept.
 - a. If changing variable mappings, be aware any changes will not be reflected until the next day.

3. If still unresolved, identify the scope of the issue. Does this affect all cases, cases within a particular time range, or just a single case?
4. Send the summary of the problem to your local technical contact. Please include any relevant information such as:
 - a. Scope of the problem
 - b. Diagnostic and/or Case Viewer screenshots
 - c. Variable IDs and names
 - d. Example MPOG Case IDs

Steps to Try (Technical Support)

1. Are there any awaiting files in the file directory?
 - a. If so, run import utility and later investigate why import utility automation is failing (see section below).
 - b. If running the import utility manually also fails and the first column of the file is a date, make sure the date format of the first column matches that of the import utility's config file. If they don't match, correct the dates in the file. You may need to reach out to the file extract developer for assistance.
2. Verify whether the data was imported.
 - a. Check the Import.Files or Import.PartialFiles tables to make sure the files for the given date exist.
3. Verify the relevant rows within the file are correct.
 - a. A file's data can be manually parsed into tabular format using the Core.ParseFileData stored procedure.
 - b. If the file is incorrect, contact the extract developer.
4. Verify whether the data was consumed.
 - a. Check the Core.Log_Consume table for the relevant Target_Date and Module.
 - b. Also check the relevant Staging table to see if the rows are present.
 - c. Running the consume process can be manually done via the stored procedure Core.ConsumeFiles.
5. Verify whether the data was handed off.
 - a. Check the Core.Log_Handoff table for the relevant Target_Date and Module.
 - b. Running the handoff process can be manually done via the Handoff.ScheduleItemsOntoQueue and the Handoff.ProcessQueue stored procedures.
6. If still unresolved, contact MPOG for additional assistance

Import Utility – Not Running Automatically

Symptoms

Files remain within the MPOG directory for longer than five minutes after being placed there.

Steps to Try (Technical Support)

1. Verify that the Scheduled Task “MPOG – Run File Import Utility” located on the MPOG application server is enabled and scheduled for every five minutes.
2. Verify the Scheduled Task refers to the correct location for the Import Utility (C:\Program Files\MPOG\MpogFlatFileImporter.exe)
3. Verify that the Windows account the Scheduled Task runs under has a login on the MPOG database server.
4. Verify that the Windows account the Scheduled Task runs under has a user account on the MPOG_Import_Manager database with the db_owner role.
5. If still unresolved, contact MPOG for additional assistance.