ISR Business Production Procedures

Last Updated on September 6, 2016
## Table of Contents

1. **Business Production website** .......................................................... 4
2. **Service Now**.................................................................................. 4
3. **About Automic** ........................................................................... 5
4. **Batch Programs** .......................................................................... 5
   4.1 Standards for Batch Programs ....................................................... 6
      4.1.1 Naming conventions ............................................................... 6
         4.1.1.1 Programs related to Banner modules .................................... 6
         4.1.1.2 Non-Banner related jobs .................................................. 6
         4.1.1.3 Data oneshot programs .................................................... 6
      4.1.2 Oracle Scripts ........................................................................ 6
         4.1.2.1 Coding standards .............................................................. 6
         4.1.2.2 Input files ....................................................................... 7
         4.1.2.3 Output files ................................................................... 7
         4.1.2.4 Variables/Prompts .......................................................... 9
         4.1.2.5 Database links ............................................................... 9
   4.2 Documentation ............................................................................. 9
      4.2.1 Job/Module Documentation .................................................. 10
      4.2.2 Chain Documentation .......................................................... 10
   4.3 Crystal Reports jobs .................................................................. 10
   4.4 SQR jobs ..................................................................................... 11
   4.5 Web Service jobs ....................................................................... 11
   4.6 mySQL jobs ................................................................................ 11
   4.7 Imaging Standards ...................................................................... 11
   4.8 Production Maintenance Roles .................................................. 12
   4.9 Web Tailor Definitions ............................................................... 12
5. **Request for Transfer and/or Setup** ........................................... 13
6. **Database Objects (Change Control Checklists)** ......................... 13
6.1 Banner Change Requests (CR's) for CCB ...............................................................14
6.2 Naming conventions..................................................................................................14
   6.2.1 Oneshot programs ...........................................................................................14
   6.2.2 Checklists .........................................................................................................15
6.3 Development (Pre-implementation) .......................................................................15
   6.3.1 Coding standards ............................................................................................15
   6.3.2 Checklist Tips ...................................................................................................16
6.4 Implementation in QA ............................................................................................17
6.5 Implementation in PROD .........................................................................................18
6.6 Change Requests – (CR, previously RFC) .............................................................18
6.7 Orange flag Items ....................................................................................................18
7 ISRBP Response Timeline ..........................................................................................19
1 Business Production website

To access the Website – follow this link http://www.mcgill.ca/business-production/

Change related requests (program transfers, checklists, maintenance roles in PROD) are all processed through Service Now. Any other requests must be submitted via the BP Request System at the site above.

If you do not have access to the BP site, make a formal request to your supervisor asking them to submit a request to grant you access – we will need your full name, McGill ID, e-mail address and the group(s) you need to belong to (SIS, FIS, etc). We will confirm once you have access.

Tips:
1. To submit a request, click on “Sign in” on top left corner and login using your McGill ID and Minerva PIN.
2. Click on Request for Service option. A drop-down menu with eight options will be displayed. Select an option and fill in the form as indicated – fields in red are mandatory – click on the submit button to submit the request – an e-mail (receipt) will be sent to you.
3. You can follow the progress of your request by going to the Request Inquiry link (on the left side menu).
4. All correspondence/interaction associated with a request must done using the “notes” feature our the BP request system – you and all parties associated with the request will receive mail (receipt/notification) – for example, you may enter a request to run a job in QA – we will run it in QA and add a note informing you that it was completed – you may then add another note (to the same request) to run the same job in production.
5. To make a copy of a request, click on “Copy Request”. Edit where needed and submit.

2 Service Now

Service Now is the ITSM tool used by McGill. It is used to create Change Requests to record changes to McGill IT infrastructure. Any change to a production system requires a Change Request in Service Now.

If you need BP to perform a task in relation to a change (program transfer, promotion, table creation or modification, etc) you will request this through a Change Task inside your Change Request in Service Now.

Tips:
1. Leave the change task state in Pending until you want us to begin working on it. When you want us to begin the work, change the state to Open.
2. Assignment group for Change tasks for BP should be ISR – Business Production
3. Create one Change Task per instance. For example, you will have one task for implementation in QA and a different task for implementation in PROD. The QA task will most likely have multiple notes back and forth while testing.
4. Be sure to provide the complete path and PVCS revision numbers for all programs, packages, documentations, etc - specify the target instance and the date required. Include a brief description or reason for all changes as well as any special instructions in the comments section.
3 About Automic

AUTOMIC is a job scheduler with which we can do all the traditional batch job management tasks, including submitting jobs and viewing output, etc...

To get access to Automic, please go to:


For support, please contact ICS.

- Module/Job is the basic building block in Automic and is created for each program. It contains all the information required to execute the program. A job can run both individually and as a component of a chain/process flow. A job can be a component of as many chains as need be. If a job definition changes, the change is applied to every chain that includes it. Every run of a job is associated with an internal sequence number that is reflected in the output file names to guarantee its uniqueness.

- Chain/Process flow is the AUTOMIC equivalent to a job stream. A chain may include one or more components (jobs and other chains), and each component execution depends on its relationship to other components in the chain. Every run of a chain has a unique chain id that is common to all its components.

- AUTOMIC can run many types of jobs (C, Banner COBOL, SQL, SQR, Pro-C, UNIX, MySQL, SQL server, web services, etc....) on many different platforms (Windows, Linux, Unix, etc.).

- AUTOMIC allows passing prompt values to jobs. The values come from fill-in prompts or substitution variables that run small SQL programs to return a value.

- Substitution variables store values that can be referenced in prompts and conditions of jobs and chains whether within the same chain or by an external job.

4 Batch Programs

To run batch jobs, Business Production uses Automic (formerly UC4/AppWorx) for job scheduling. A brief description of AUTOMIC environment followed by the requirements and standards for transferring programs, input files and documentation from Development to QA and Production follow.
4.1 Standards for Batch Programs

4.1.1 Naming conventions

4.1.1.1 Programs related to Banner modules

Banner-supplied programs: jobs set up in Automic will have the same name as the Banner program.
E.g. FGRACTG, TGRAPPL, SFPFAUD, etc... where the first letter denote the system – F for Finance, T for AR, S for Student, etc...

McGill-created jobs: will resemble the banner naming convention except the second letter will be a Z to denote it’s an in-house developed program. And the length should not exceed 8 characters. E.g. szpacde: S for Student, Z for McGill job, P for process. Note for the 3rd character the most often used are P for process and R for report.

Note for webservice jobs, use W as the third character.

4.1.1.2 Non-Banner related jobs

BP can set up and run jobs against other systems, such as Research, Lenel, Architecture, etc...

The first character is always a Z. The 2nd character denotes the system while the 3rd is P for Process or R for Report. E.g. ZAPIMPO: Z for McGill, A for Architecture system, P for Process.

It’s always recommended to talk to a BP member about your names before creating and saving your files.

4.1.1.3 Data oneshot programs

The name must follow the Banner standard in positions 1 and 2. Position 3 to 5 must contain ONS and position 6 to 8 must contain a number which is incremented with each new request/version.  i.e.: pzons999 for Payroll oneshots, fzons999 for Finance oneshots, etc... Note: All program names should be in lower case.

Oneshots need to be accompanied by a module doc such as fzons999_module.doc, even if part of a checklist (see section 6.2.1 for checklist oneshots).

4.1.2 Oracle Scripts

4.1.2.1 Coding standards

1. If processing is to stop at the first error encountered, start the script with a whenever sqlerror &&so_exit . &&so_exit is an environment variable set by BP to EXIT FAILURE.
2. If processing is to continue despite the failure of a statement you need to toggle back and forth between `whenever sqlerror continue` and `whenever sqlerror &&so_exit`. Place `whenever sqlerror continue` before the statement(s) that is allowed to fail and `whenever sqlerror &&so_exit` before the statement(s) that will prompt the script to stop and exit with a failure.

3. A commit is implied in EXIT FAILURE. So if you’d like to roll back use `whenever sqlerror exit failure rollback`. This is especially true in the case of data oneshots and multiple SQL statements.

4. If your code is not a oneshot and you still want to rollback in case of error, you can do the following:

   `DEFINE exit_subst = 'EXIT FAILURE ROLLBACK'
   WHENEVER SQLERROR &&exit_subst;`

5. In PL/SQL blocks use the RAISE_APPLICATION ERROR function to re-raise an exception that has been handled so that the PL/SQL block will now end with an ORA error, which will cause SQLPLUS™ to abend because of the WHENEVER statement.

6. Always, add a slash (/) at the end of an anonymous block. But do not add a slash at the end of a sqlplus script.

### 4.1.2.2 Input files

- The naming convention of Input data files must be discussed with BP prior to the first request.
- Input data files must be place on the ftp secure server in the appropriate directory. Access to the secure server can be granted when requested from BP.
- The path for the input file name must be replaced with `&&so_inpath` in the script.
- The file name should be replaced with a parameter such as `&&so_infile`. This parameter will be passed in the job set up.

### 4.1.2.3 Output files

#### 4.1.2.3.1 File extensions – lis, dat, csv, slk:

To create lis files use the following variables:

   `&&so_outfile to &&so_outfile35` - The maximum allowed is 35 files.

   Note that the first output file does not use the suffix of “1”

To create data files use the following variables:

   `&&so_outdat1 to &&so_outdat20` - the maximum allowed is 20 files.

To create csv output use the following variables:

   `&&so_outcsv1 to &&so_outcsv20` - the maximum allowed is 20 files.

To create slk output use the following variables:

   `&&so_outslk1 to &&so_outslk20` - the maximum allowed is 20 files.
4.1.2.3.2 Changing output file names:

Output files produced by AUTOMIC have the following nomenclature:
{module}_J{chain_id}-j{seq_no}.ext.

If you need to change the name of the output file generated by an SQL, there exist two specific functions for this purpose: f_return_modified_filename & f_tilde_modified_filename. Both exist in package GZKUTIL.

The concept entails inserting a string in the file name without changing the original structure. The string will be inserted after {module} – which is a reserved word in AUTOMIC and is replaced by the job name at run time (alias).
To precede the string by an underscore, use f_return_modified_filename.
To precede the string by a tilde (important for Minerva), use f_tilde_modified_filename.

An example of using f_return_modified_filename:

Script szrgdrj has run as part of a chain and the output name is SZRGDRJ_J12345678-j12345678f1.csv
The word “report” will be inserted between the job name SZRGDRJ and the chain id.

Variables needed to call f_return_modified_filename

```
V_FILE_NAME       VARCHAR2(100)  := ' ';  v_repl_str         VARCHAR2(50)  := ' ';    v_new_file_name VARCHAR2(150);
```

In the declaration section of the script, variables to handle the output filename are declared.

```
rej_filename            varchar2(60) := '&&so_outcsv1';
```

```
procedure
open_utl_file is
begin
v_file_name gets set to whatever AUTOMIC has passed to the script.
v_file_name    := rej_filename;
Set up the variable you wish to use to modify the AUTOMIC name passed to the script. (eg report)
v_repl_str     := v_term||'_'||'report';

Call package gzkutil and specify the function return_modified_filename while passing your variables.
rej_filename   := gzkutil.f_return_modified_filename(v_file_name,v_repl_str);

What you get back from the function is a combination of the two variables passed:
ISR BUSINESS PRODUCTION PROCEDURES

is SZRGDRJ_REPORT_J12345678-j12345678f1.csv

Note: Failure to use the above procedure may result in errors registering the output files by Automic.

4.1.2.3.3 Output files on Minerva

- Every effort must be made to post output destined for the users, on Minerva. For that, the object file must be created by the requester and processed by BP. And the Minerva security designate will assign the created objects to the relevant classes.
- In the case of exception reports, AUTOMIC can email the file provided the data is not confidential. If the data is confidential, the file must be posted on Minerva, and AUTOMIC will email a notification to the user alerting her to the existence of output on Minerva.
- Please ensure that all BP staff have access to the new report(s).

4.1.2.4 Variables/Prompts

AUTOMIC also allows prompt values to be passed to the program prior to submission. AUTOMIC replaces words preceded by & and/or &&. The variable name in the SQL script must be documented in the AUTOMIC module document.

Example:

```sql
DECLARE
    v_term_code      VARCHAR2 (6)   := '&&term_code';
BEGIN
    BANINST1.SZKCAVG.PM_CALC_CLASS_AVG( v_term_code );
END;
/
```

&term_code must be documented in the prompts section and added to the prompt set up in AUTOMIC job.

Note: if the value of the prompt includes an ampersand, replace it with ‘| |Chr(38)| |’.

4.1.2.5 Database links

- Database links should not be embedded/hard-coded in your scripts.
- Please replace all occurrences of the database names (ex. @BANNER) with @&&db_link. Then add a new prompt to the appropriate field in the module documentation. Thus the value for the prompt passed by AUTOMIC should only be the instance name e.g. BANNER.

4.2 Documentation

All programs and procedures run by AUTOMIC must be documented as follows:
4.2.1 Job/Module Documentation

• Documentation must be based on the template in PVCS at OPERATIONS/Documentation/(Template)
  Batch Job Setup Form.doc.
• Documentation should conform to the following: modulename_module.doc.
• When transferring a new version of an already existing one, please state the changes in the notes
  section in the Change Task and in the new document, preferably by making new text bold and by
  striking out old text or using the highlighter.

4.2.2 Chain Documentation

• The naming convention for the chain/process flow document title is system id-chain
  name_chain.doc.
• System ID’s or AUTOMIC application name are: AR, GEN, DWH, FIS, etc, followed by a dash, the
  name of the chain and “_chain.doc”. e.g.: DWH-DataWarehouse_Load_Chain.doc

In the chain documentation, the following details must be included:
  - Time, day, and frequency of chain execution – including run instructions during holidays
    (Summer Fridays, Christmas, etc…) – Note that some critical jobs may require Developers or
    Analyst to be on-call off hours so please be aware of this.
  - A list of relevant parameter values for each job.
  - The inter-dependency among the jobs constituting the chain, i.e. any sort of
    dependency(success/failure) on the other modules with the chain.
  - If applicable, list other conflicting modules and/or chains, which are not to run while the
    chain in question is running.
  - Relevant error handling procedures. For example: if step5 aborts, rerun starting with step 2,
  - Importance of job failure resolution – immediate notification or can wait until next working
    day.

4.3 Crystal Reports jobs

• Crystal reports transferred to Business Production should be production scripts only.
• Output will not be emailed and and passwords should never be embedded in the reports.
• When saving a Crystal Report be sure not to save data with the report. Click on File and uncheck “Save
  data with Report” before saving.
• In the module documentation, please fill in the fields Appworx/Crystal Report Output format to specify
  the type of output file to be produced:
XLS: Formatted Excel
XLSRECORD: Unformatted Excel
PDF: PDF

4.4 **SQR jobs**

In the module documentation, please fill in the fields *Appworx/SQR Report Output format* to specify the type of output file to be produced:

EP: HTML & PDF Files
EH: HTML Files
PD: PDF or TXT Files
PS: Postscript
EPC: Excel, HTML and PDF Files
EHC: Excel and/or HTML Files
SPF: SQR format

4.5 **Web Service jobs**

Before BP is able to set up web service jobs an adapter/jar file has to be built from a wsdl file with the url, the security policy has to be determined and a login has to be created with the Web Service URL.

4.6 **MySQL jobs**

Create a .bat script that will write out a temp SQL %mysql_script% using “echo” command. The %mysql_script% temp file will be dynamic and will be handled by Automic MySQL wrapper.

The required variables (ex: database name) will be passed by Automic prompts and will be replaced when the temp sql is executed. Thereafter, Automic will clean up the temp file and any zero size output. Automic will also save the temp sql contents in the log file.

4.7 **Imaging Standards**

Please follow these standards when transferring jobs that produce image files, and which need to be loaded onto the imaging server.

- Image output files should have the following string “~IMAGE” immediately following the program name.
- A text file with the name: {module}~IMAGE_J{chain_id}-j{seq_no}.txt should be produced by the script.
- This text file lists all image filenames with the path on imaging server, along with file_specific data such as student ID.
- A text file with the name: {module}~TOTALS_J{chain_id}-j{seq_no}.txt is to be produced by the script. This file indicates the total number of files/records.
• Image application name is to be provided, such as B-S-ADMIN.
• In cases where the image-creating module runs more than once within the same chain, a unique identifier must be provided for within the name of the files for each run – usually dependent on the prompts used per run.

4.8 Production Maintenance Roles

If you require maintenance rights to QA, please submit a web request (Request maintenance roles).

Manager approval is required for maintenance rights in QA. Ensure that you include the email address of your Team Lead in the CC field box on your web request so that they can add a note stating their approval.

Maintenance rights in production require a CR to be created. Once the CR is created, submit a Change Task to ISR – BUSINESS PRODUCTION requesting the maintenance role. Access will be given once the CR is approved.

When requesting maintenance rights, be sure to include the following information in your request:

• A short 1-2 line description of the work you are doing
• Any applicable HEAT ticket numbers you are resolving
• The name of any tables being worked on
• SQL statements being run, when applicable

You will be informed once the role has been granted and once it has been revoked.

4.9 Web Tailor Definitions

A Web package will contain a number of procedures that generate web pages. Any Web page that is dynamically built as a result of a mouse-click is generated by a specific procedure within a package. Each procedure within the package is given a unique name, and each such procedure must be “registered” with Web Tailor in order for it to become accessible through the Web.

The Web Tailor definitions of interest (i.e. the information captured when procedures and menus are registered/defined) are stored in the following tables (WTAILOR schema):

1. TWGBWMNU – basic procedure and menu information
2. TWGRINFO – information text associated with procedures and menus
3. TWGRMENU – menu items for each menu (can also be attached to procedures)
4. TWGRWMRL – user roles for which procedures and menus are enabled

As an extension to the above 4 tables, any Web Tailor information which has been modified/translated for the French Web site is stored in a 5th table (WEB1MGR schema):

5. NLS_LANG_TEXT – modified/translated text for multilingual support

WebTailor definitions are migrated between database instances using scripts. Developers can write their own oneshot scripts or can request BP to run:
ozowtcopy.sql - to copy the rows of a specific procedure within a range of activity dates from one database to another. ozowtdel.sql - to delete the rows of a specific procedure from the database. ozowtnfo.sql – to copy a new row in twgrinfo by label name for an already existing procedure. ozowtinfd.sql – to delete a row by label name for an existing procedure.

The above jobs with their required parameters are on the checklist template.

Important: Migration of menu items should be discussed with ICS prior to requesting the scripts from BP. Formal ICS approval at the QA stage is required for all significant changes before they can be migrated to production.

5 Request for Transfer and/or Setup

1. Pre-transfer:
   a. QA:
      Move your source code (Programs, Shell scripts, SQL Loader, control files, etc), and documentation to PVCS:
      If applicable, move your input data files into the relevant Input Data folder, i.e. Test_FIS_Input_Data, Test_SIS_Input_Data, etc...
   b. Production:
      Unless the documentation is changing, you need not provide another one.
      Move your source code (Programs, Shell scripts, SQL Loader control files, etc), and documentation into PVCS
      Move your input data files into the relevant Input Data folder, i.e. FIS_Input_Data, SIS_Input_Data, etc...

2. Submit a Change Task to ISR – BUSINESS PRODUCTION through your CR in Service Now. Submit one task per instance.

3. Important – All requests for program transfers to Production require approval from the Team Leads. The transfer will not be done unless the Change Request (CR) is approved in Service Now.

6 Database Objects (Change Control Checklists)

This outlines the main steps required to create and/or modify database objects such as tables, packages, functions, etc... from Development to QA and Production starting with creating the CCB Change Request to submitting the Service Now Change Request (CR).
6.1 Banner Change Requests for CCB

This section outlines the steps/flow required to create and process a Banner Change Request to be reviewed by the Change Control Board (CCB).

1. An analyst in one of the portfolio teams, in conjunction with the functional users, creates the change request (CR) document completing the required information.

2. The Portfolio Manager or designate assigns a change request identification number according to the following standard, Team_YYYMMDDa,b,c etc. (e.g. sis_20130228a), and ensures the correct completion of the CR.

3. The CR should have a MOD number. If a new MOD needs to be created, MOD_template.docx can be accessed on BP site. Once filled out and given a proper name and deposited in \file\staff\Banner\Projwork\0_Change_Management\MODS, a change task is to be submitted to create the new MOD.

4. The Portfolio Manager or designate deposits the change request document in SharePoint site at https://agora.mcgill.ca/isr/ccb/). There is a separate change request folder for each calendar year.

5. The Portfolio Manager or designate submits a CR in Service Now to seek approval for the CR. There should be a change task assigned to the CCB assignment group. The task should have a link to the specific CR folder on the Change Control Board SharePoint site (see sample change task below).

6. BP logs all Banner change requests in the ZARK database. The information stored is taken directly from the change request form set up in SHAREPOINT. The change request record created in ZARK uses the same naming convention (e.g. sis_20130228a). You can see an example of a Banner change request in CTASK0011212 in Service Now.

7. Meanwhile, Sue Reali will receive a copy of the change task, go to the CCB SharePoint site to review the CR stored there, and add a note to the change task stating whether to approve the Banner CR or to send it for further review to the Change Control Board (CCB).

8. If you or any members of your staff who will deal with Banner change requests need access to the Change Control Board SharePoint site please submit an ISRBP web request (type 5) indicating who should be given access.

Note: Banner checklists will not be processed in any database unless the underlying Banner CR has been approved.

6.2 Naming conventions

6.2.1 Oneshot programs

All statements that are related to a checklist need to be scripted into oneshot programs.
The CC naming convention described below is only used for oneshots that modify database structure. If you need to run a data oneshot as part of your checklist you will follow the same naming convention outlined in section 3.1.1.3 (ex SZONS715). Data oneshots as part of a checklist still require a module doc.

The naming convention follows the Banner standard in positions 1 and 2. Position 3 and 4 must be CC and position 5 to 8 must contain a number which is incremented with each new request/version. i.e.: PZCC9999 for Pyaroll, FZCC9999 for Finance, etc... Note that there's no need to create a module doc for oneshots that are part of checklists.

6.2.2 Checklists

For every new checklist please pick up the template from this location. Do not copy older checklists so that you always have the most recent template.

https://home.mcgill.ca/files/business-production/Checklist_Sample.docx

Give the checklist a file name using the following format:

YYYYMMDD_<System>_short_description_of_request.docx

where YYYYMMDD is the submission date and <System> such as ‘SIS’, ‘FIS’, etc...

6.3 Development (Pre-implementation)

6.3.1 Coding standards

1. All scripts and database objects should be tested in SQL*Plus prior to filing a request with BP. All PL/SQL programs should finish with a / (slash). This is what triggers the block to execute in an sqlplus environment.

2. If processing is to continue despite the failure of a statement you need to toggle back and forth between “whenever sqlerror continue” and “whenever sqlerror exit failure”. Place “whenever sqlerror continue” before the statement(s) that is allowed to fail and “whenever sqlerror exit failure” before the statement(s) that will prompt the script to stop and exit with a failure.

Beware the failed statement will generate an Oracle error but it will be ignored.

3. If it is certain that a particular statement will fail, it’s best to comment it out if toggling between continue and exit failure is not desired.
4. **Create table scripts:**
   - Ensure that you have obtained the required approvals for a new table, and indicate so in the approvals section of the checklist. Table approvals come from CCB (Sue Reali).
   - Ensure that you precede the table name with the schema.
   - Ensure that you have scripted for the grants that are appropriate in QA and Production – which is not necessarily that of Dev.
   - Tablespace may default to USERS. Please ensure that the tablespace for a table is BANDATA and for an index is BANINDX.
   - The blank spaces that PVCS generates while adding the comments causes a failure in running the script an sqlplus environment. If you like to keep the keywords in PVCS, start with the statement “set sqlblanklines on”.

5. **Create View scripts:** It’s important that the view name be preceded by the schema name.
   To keep the keywords in PVCS, start with “set sqlblanklines on” so that sqlplus ignores the blank lines within the PVCS keywords, followed by the column names, followed by the Select, followed by pvc keywords.
   Note: If ever you copy the script from toad to update it, it won’t have the set blanklines on because it’s outside the select.

6. **Grants:** While compiling new objects (packages, procedures, functions, etc..) or new revisions to existing objects, BP via the “compile” chain applies standard execute grants on said objects to ALUMNI, BANIMGR, BSACIMGR, FAISMG, FIMSMGR, GENERAL, PAYROLL, POSNCTL, SATURN, TAISMGR, WTAILOR, BAN_DEFAULT_M and BAN_DEFAULT_Q.
   If the package being compiled starts with a BZ, BP will also grant execute on the package to WWW_USER408 and WEB1USR.
   Scripts must be created for any other required grants.

7. **Synonyms:** While compiling new objects (packages, procedures, functions, etc..) or new revisions to existing objects, BP via the “compile” chain will create public synonyms.

### 6.3.2 Checklist Tips

1. When modifying tables indicate whether the data warehouse would be affected by the change and inform the data warehouse team of the planned change. To find out if there’s an effect on the data warehouse, you can run the code below from Banner
   SELECT *
   FROM dwh.gz_dwh_replicated_objects
   WHERE name like UPPER('&object_name');
   Or you can ask BP to run job OZRDWOBJ which returns the results of the sql above.
2. The order of execution of the different components of the checklist may be very important to avoid unnecessary failures and delays. If this is the case with your checklist please indicate so and label the steps in the order required.

3. Please consult the Web Programming Standards document when creating or modifying Minerva Web pages.

4. If the system being developed or changed needs English/French translation, indicate so on the checklist so that the objects are compiled in the standard schemas as well as in web1mgr.

5. Where required start working with the ICS Training & Documentation group to create documentation, help screens, training courses, and announcements and ensure conformity with the McGill “look-and-feel” standard as soon as possible.

### 6.4 Implementation in QA

1. Save the checklist in the folder:
   \file\staff\Deptshare1\Banner\Projwork\0_Change_Management\Implementation_Checklists\New

2. Submit a Change Task and assign it to ISR – BUSINESS PRODUCTION. The task will not be worked on until it is in the ‘Open’ state. There should be only one task created per instance.

3. Your checklist will be moved to the “In Progress” folder, located at:
   \file\staff\Deptshare1\Banner\Projwork\0_Change_Management\Implementation_Checklists\In Progress

   When BP implements the checklist, BP updates it by checking off the appropriate checkboxes on the left-hand side of the form and by adding any other pertinent information that would be meaningful for the implementation in Production.

4. If problems arise in QA testing, Do NOT start a new checklist or give it a new name, but rather update the existing one to reflect changes, then add a note to the same Change Task while specifying that it is a re-submission to QA and details what items need to be re-implemented.

5. The QA implementation process helps to validate the checklist for accuracy and completeness. If problems arise, the checklist must be modified accordingly. Please note that the checklist is a “live” document and all involved are encouraged to enter as much information as possible on the checklist during the QA phase. Any notes that could help clarify the implementation into Production should be entered into the appropriate areas.
6.5 Implementation in PROD

1. Important – All requests for production implementations require approval from the Team leads.

2. All implementations in PROD must be within the production window on Wednesdays between 2pm – 6pm. Banner changes are from 4pm-6pm.

3. Migration to Production must be explicitly requested by adding a new Change Task to the previously created CR indicating that the checklist should now be migrated to PROD. Do NOT create a new CR for migration to Production. Do NOT create a new checklist for migration to Production. Always use the same CR and checklist that was used for the migration to QA.

4. Migration to TBAN3 (the training database) is performed if the corresponding box is checked in the checklist. It may also be necessary to compile in other instances for upgrades or other special situations.

5. When the checklist has been completed, it will be moved to the “Completed” folder, located at: \file\staff\Deptshare1\Banner\Projwork\0_Change_Management\Implementation_Checklists\Completed\A note will be added to the change task and the task will be closed. The CR should be closed shortly after

6.6 Change Requests – (CR, previously RFC)

Every change in production requires a CR to be created in Service Now (we previously used RFCs that were created in IT Service Motion).

If you cannot login please ask your supervisor to get you access.

Once BP informs the requester that the implementation has been completed, it is the requester’s responsibility to close the CR after completing all required fields.

6.7 Orange flag Items

BP maintains a list of known sensitive and critical objects. Compiling any of these packages in production requires Banner downtime, unless otherwise approved by the EEC.

The following objects are considered orange flag, or problem packages: GZKUTIL, SZKmutt, SZKUTIL, SZKCONS, TWBKWBIS, GS_NLS, GS_DATE_PKG, BGKHTML, TWBKLOGN, EFKCUST.

In addition, the package bodies HZSKTAIL.PKB and HZSKALOG.PKB compile cleanly but as they are used to access the web, their compilation can be disruptive to current sessions causing 00600 messages to be generated and requiring the DBA’s to step in and ‘restart’ the web servers. The disruption occurred during a high web usage period when students were accessing the system in great numbers.
7 ISRBP Response Timeline

**Hours of Operation** - *The following times are to be used as a guideline, we will try to respond to your request on a “best effort basis”*.  

08:00 to 17:00 workdays

Simple requests will be processed asap between 08:00 to 17:00 (program/documentation transfers, running modules/chains, etc). Jobs can still be scheduled to run outside these hours.

Allow at least 48 hours for complex requests.