

# tranSMART 19.0 Release (Aug 2019)

Release 19.0 (August 2019) is a **major update** to the previous release 16.3.

## Release Notes tranSMART 19.0

### Version Number

This version is renumbered to reflect the year of release, and to indicate the major effort in rewriting and reorganizing code.

This is also the first official tranSMART release to be compatible with the parallel i2b2-transmart project. The intention is for i2b2-transmart to use this release, perhaps with a small number of additions to integrate with their latest changes in other codebases.

### Test Version

A test version of the full release is available at <http://postgres-test.transmartfoundation.org/transmart>

Details of the beta test server data and of the features to be tested are in the [beta test public instance wiki page](#)

### Single repository

TranSMART 19.0 code is reorganized in a single repository <https://github.com/tranSMART-Foundation/transmart>

The top level directories are merged copies of the tranSMART 16.3 repositories with minor changes.

The combined repository makes release branches simple to manage. A single branch in the main repository can be used to generate all the artifacts for a release distribution.

The directories mirror the original organization of the source code for the tranSMART 17 project to simplify direct comparisons,

### transmart-core-db and transmart-core-db-tests

The test code previously under transmart-core-db/transmart-core-db-tests has been relocated to its own top-level directory.

This makes building and testing simpler, and was also the organization chosen by the transmart 17 server-only project.

### transmart-etl

The release 16.3 tranSMART-ETL repository has been renamed to all lower case. Unused legacy directories have been purged from the repository, greatly reducing the size of the zip file generated for each release.

- FCL4tranSMART is superseded by transmart-ICE which has the latest released code from Sanofi for their ICE tool data loader (formerly FCL4tranSMART).
- The V1.2\_Hackathon directory contents were unused
- The loader is now superseded by the coipuy built from the src directory
- The src-old directory is clearly redundant
- The duplicate Kettle scripts under Kettle-GPL, Oracle and Postgres are removed. Since the start of the release 16 series the release copy of these scripts has been in the database-specific directories under the Kettle directory. These are used by the make targets in transmart-data and should be used by any local ETL pipelines.

### transmart-extensions

The transmart-extensions plugin in release 16.3 has been split into its three components:

- transmart-java
- biomed-domain
- search-domain

This also reflects the rearrangement in the tranSMART 17 project.

### Galaxy-export

The old release 16.x 'blend4-plugin' is renamed galaxy-export-plugin.

Throughout the code the name 'blend4j' has been replaced to make the functionality clear.

### SmartR

SmartR was developed by the eTRIKS project and released in tranSMART 16.2 with a set of interactive analysis workflows that supersede many of the functions of the "Advanced Workflows" tab.

We are testing new SmartR workflows developed by other partners in the eTRIKS project to provide the remainder of the "Advanced Workflows" functionality.

The Advanced Workflows remain active in this release. We anticipate that users will require them in order to reproduce previous analysis, and they can be used to compare results and encourage migration to SmartR.

## Fractalis

Fractalis was developed for i2b2-tranSMART by the same author as SmartR (Sasha Herzinger at the University of Luxembourg) and superseded several of the SmartR workflows.

We are working on the full integration of Fractalis into tranSMART 19.0.

## Database schemas

### tranSMART and i2b2

The database schema has been updated to resolve, as far as possible, differences with the i2b2 1.7.12 schemas.

Where columns are date or time values in i2b2 and string values in tranSMART, they have been corrected to the appropriate date or time values. Initial testing found no conflicts in ETL procedures.

Required columns are updated in tranSMART so that any column required by i2b2 in a common table are also required in tranSMART. One date needed to be defined in ETL procedures using an obvious default value. No other impacts have been noted in initial testing.

Triggers are required for some tables in i2b2. Although PostgreSQL can be configured automatically to increment unique identifiers for new rows in a table, the i2b2 code may include a call to increment a named sequence to generate a unique id value when a row is inserted. This necessitates defining a sequence with this name and using the same sequence in a trigger function to maintain database integrity.

Integers in PostgreSQL are defined as type 'int' unless extremely large values are expected. Values that can exceed 1 billion are defined as type 'bigint'.

### New schemas

Four additional i2b2 schemas have been included:

- i2b2hive
- i2b2imdata
- i2b2pm
- i2b2workdata

### New i2b2 Tablespaces

i2b2 does not define tablespaces by default. A set of tablespace names were agreed with the i2b2-tranSMART developers for implementation in the tranSMART schema. All i2b2 tables are in tablespace I2B2 and all i2b2 indices are in tablespace I2B2\_INDEX. These reflect the tablespaces TRANSMART and INDX used for the tranSMART-specific tables and indexes.

### Redundant tranSMART tablespaces

Earlier tranSMART releases defined 3 additional tablespaces: biomart, deapp and search\_app. These were no longer used - though they were created with a new database. They have been removed from tranSMART 19.0.

For this release they are ignored by tranSMART code

New tables have been added to the schemas used by tranSMART 16.3

Support for tranSMART\_batch on Oracle is included by default. In tranSMART 16.3 this had to be installed before tranSMART\_batch could be used with an Oracle database.

We are working on including support for tMDataloader on Oracle and PostgreSQL by default. This involves running the respective installation scripts and incorporating the changes into the PostgreSQL and Oracle schema definitions, including the tMDataloader specific versions of ETL functions.

### Default passwords for database roles

The usernames and passwords for the database roles are defined for Kettle when the database is created. By default the username and password are the same as the schema. While this is simple for developers, production instances should define unique passwords for each role. By adding these as environment variables when launching the database creation target the Kettle properties files will be populated. We recommend retaining the role names as these are used in many places in the code.

### Default passwords for tranSMART login

A limited set of usernames is defined for this release, as for previous releases. These have default password, usually 'transmart2016' except for username 'admin' with default password 'admin'. These are simple for developers. Productions instances should change the passwords using the 'Admin' tab to manage the available user accounts.

Notwithstanding this, there is also a configuration option (turned off by default) to enable a guest login that by default will allow a visitor to a transmart server to login as username 'guest'. If using this option, changing to another username requires logout through the 'Utilities' menu, or explicitly using the login page URL.

## Initial database content

A new database (for Postgres or Oracle) is defined as a target in transmart-data.

A common set of data is loaded for both databases. This includes a set of standard ontologies (or data dictionaries). In this release these are:

Data	Source	Date	Table
Human genes	NCBI		biomart.bio_marker
Mouse genes	NCBI		biomart.
Diseases	Medline		biomart.
Countries		Selected	biomart.
Therapeutic Areas		Selected	biomart.
Organisms	NCBI Taxonomy	Selected	biomart.
Platforms	GEO etc.	Selected	biomart.
Human Pathways	HMDB		biomart.
Pathways	KEGG	Last public release	biomart.
Proteins	SwissProt/UniProt		biomart.
MicroRNAs	MiRBase		biomart.

The data from these sources is included in the searchapp.search\_keywords table with synonyms in the searchapp.search\_keyword\_terms table.

## Grails upgrade

### Grails version 2.5.4

This release is built using Grails 2.5.4.

Grails can be installed using:

```
sdk install grails 2.5.4
```

Earlier releases up to 16.3 were built using Grails 2.3.11 and Grails 2.3.7

### Building individual components

Each directory has a build script.

Grails components use the `./grails` script with the targets `package-plugin` (or `packagePlugin`) and `maven-install` (or `mavenInstall`)

Maven builds use the `./gradlew` script with targets `clean`, `build` and `publishToMavenLocal`

In two component where these scripts are not yet built, the build used maven directly.

For two external packages `mydas` and `lpaApi`, `cd` to the directory and build with:

```
mvn install
```

### Component dependencies

The recommended build order to ensure dependencies are satisfied by previously built components is:

Directory	Name	Dependencies	Description
transmart-core-api	transmart-core-api	-	Core API

transmart-shared	transmart-shared	-	<b>New in 19.0.</b> Provides generic utilities to return security information about the current user to remove the need to pass the username around in service calls.
transmart-legacy-db	transmart-legacy-db	-	
transmart-fractalis	transmart-fractalis	-	<b>New in 19.0.</b> Adds the Fractalis interactive workflow tab, integration is a work in progress.
mydas	mydas	-	The original DAS code imported into tranSMART because there is no guarantee the original distribution will remain available.  A small section of code is intended to be autogenerated, but this is easy to maintain by hand if any of the dependencies change.
dalliance	dalliance	-	The dalliance genome browser.
transmart-core-db	<b>transmart-core</b>	transmart-core-api transmart-shared	
transmart-core-db-tests	transmart-core-db-tests	transmart-core-api transmart-core	<b>Moved to a new directory for 19.0.</b>  Tests for the transmart-core-db code, also used by SmartR and transmartApp.
transmart-mydas	transmart-mydas	transmart-core-api mydas	
transmart-java	transmart-java	-	<b>Moved to a new directory for 19.0.</b>
biomart-domain	biomart-domain	transmart-java	<b>Moved to a new directory for 19.0.</b> Domain definitions for the large number of tables in the biomart schema.
search-domain	search-domain	transmart-shared biomart-domain	<b>Moved to a new directory for 19.0.</b>  Domain definitions for tables in the searchapp schema.  This covers keyword searching and user, role and access management as both are defined in the searchapp schema.
transmart-rest-api	transmart-rest-api	transmart-core-api transmart-shared transmart-core transmart-core-db-tests	
transmart-custom	transmart-custom	transmart-shared search-domain	<b>New in 19.0.</b> Provides services to customize aspects of the user interface using new application tables.  Documentation is needed for these new capabilities.
folder-management-plugin	<b>folder-management</b>	transmart-core search-domain	
Rmodules	<b>rdc-rmodules</b>	transmart-core-api transmart-shared	Provides all the Advanced Workflows.  There is also a dependency on data loaded into the searchapp schema to define the inputs, outputs, parameters and scripts for each workflow and to define their names and the order in which they are presented. This was intended in the pre-open source versions of tranSMART to allow users/administrators to edit these settings, but this makes little sense to control fixed scripts distributed in the transmart.war file. There is no known instance of a site developing their own Advanced Workflows through these mechanisms.
spring-security-auth0	spring-security-auth0	transmart-shared transmart-core search-domain transmart-custom	<b>New in 19.0.</b> A new Auth0 controller and services.  Documentation is needed for these new capabilities.
lpaApi	lpaApi	-	This code provides one third-party SmartR workflow to interface to Ingenuity Pathway Analysis. SmartR includes hooks to load the lpaApi workflow
SmartR	<b>smart-r</b>	transmart-core-api transmart-core transmart-core-db-tests lpaApi	<b>Several potential new workflows from the eTRIKS project are candidates for inclusion.</b>
galaxy-export-plugin	galaxy-export-plugin	transmart-shared transmart-legacy-db rdc-rmodules	<b>Directory renamed from blend4j-plugin for 19.0.</b>  A version of data export that transfers to an instance of Galaxy for further analysis.  The user needs credentials to use the galaxy instance, defined in the server Config.groovy file.

transmart-metacore-plugin	transmart-metacore-plugin	transmart-shared	
transmart-xnat-viewer	<b>xnat-viewer</b>	transmart-core-api search-domain	
transmart-xnat-importer-plugin	<b>transmart-xnat-importer</b>	transmart-shared biomart-domain	
transmart-gwas-plugin	<b>transmart-gwas</b>	transmart-shared transmart-legacy-db transmart-core search-domain rdc-rmodules folder-management	
transmart-gwas-plink	transmart-gwas-plink	rdc-rmodules	
transmartApp	<b>transmart.war</b>	transmart-fractalis dalliance transmart-core-db-tests (test) transmart-mydas transmart-rest-api spring-security-auth0 smart-r galaxy-export-plugin transmart-metacore-plugin xnat-viewer transmart-xnat-importer transmart-gwas transmart-gwas-plink	This is the full transmart server, providing all the functions of the User Interface plus the access methods for the RESTful API to generate and serve authentication tokens and to serve results when these tokens are presented.

## Java version

Grails 2.5.4 uses Java 8.

Development is using the openjdk java 8. We will confirm later the suitability of oracle java 8 which is only available from third-party sources for the Ubuntu 18 test systems.

There is no longer a need for a legacy Java 7 install to work on tranSMART development.

## Asset Pipeline

Javascript, stylesheets, and other resources are packaged and provided through the [asset-pipeline](#) plugin.

This involves a major reorganization of the source files and changes to hardcoded file paths.

This upgrade is a major step towards preparing for an upgrade to Grails 3 or the newly released Grails 4 in a future tranSMART release

## Code review

A major code review was conducted by Burt Beckwith at Harvard as part of the inclusion of tranSMART 16.x code in the i2b2-transmart project.

The planned changes were described in the [i2b2-transmart roadmap](#).

## Summary of tranSMART 16-2 changes in i2b2/TM 18.1-beta

### optional support for Auth0 authentication

A new directory spring-security-auth0 provides Auth0 services.

### Groovy code formatting and consistency

Coding standards have been applied to groovy code:

- leading white space
- code indentation
- positioning of braces:
  - if/else blocks: newline after {
  - } on new line
- splitting of long lines
- Map and List values on single line is possible
- replace -each for a Map or List by a loop with a datatype and named variable specified for the value
- Use boolean for all true/false variables
- Use int for integer variables
- Use single quotes for string values except to avoid escaping many single quotes for readability
- Use {} to insert values into strings

## **updated logging to use Slf4j wrapper and parameterized logging for performance and to help with Grails 3 Logback migration**

All source files that used Log4j and calls to log.info:

- import groovy.util.logging.Slf4j
- define Slf4j as 'logger'
- call logger.info (etc.) to write to log output

## **converted most Config lookups to use Spring's @Value annotation**

Replace configuration parameter references with @Value definitions using with org.springframework.beans.factory.annotation.Value

## **use "private @Autowired" for dependency injection to reduce API pollution**

Add @Autowired references with org.springframework.beans.factory.annotation.Autowired

## **replaced many uses of 'def' with actual types, particularly in method signatures**

Throughout the code of release 16.3 types were undefined. Adding explicit types wherever possible provides validation of the type sactually passed and improves the usefulness of error messages when code breaks.

## **converted many cases of copy/paste to use methods**

A single method can replace a set of identical code segments making testing and maintenance far more robust.

## **domain class cleanup, removed many unnecessary declarations**

Domain classes have been reviewed and matched to the updates database schemas.

## **simplify access to current username, user id, roles, etc. via new SecurityService**

The new SecurityService in transmart-shared provides calls to return information on users and roles for implementing access polcies.

## **removed passing AuthUser to methods that always work on the current authenticated user, moving the lookup to where it's used**

A new directory transmart-shared provides utility functions. These include generic checks for the capabilities of the authenticated user, allowing the username to be removed from many method calls where it was being passed down.

## **upgraded to Grails 2.5.4, Servlet API 3, Java 8**

Release 19.0 is built using grails 2.5.4 which depends only on java 8.

## **tests cleanup, converted to Spock**

Tests updates in transart-core-db-tests

One test currently fails. It is testing something that is supposed to fail, but should be trapping the error condition and reposting a test success.

## **converted controller closures to methods**

Closures are defined as methods with a set of parameters, replacing the closures and parameter fetching in earlier releases.

Much of the code remains unchanged within the method aside for parameter handling and other coding standard changes (see above)

## **controller params simplifications**

Parameters are explicitly defined in each method

This give cleaner code where it is obvious what parameters are used and what parameters are available to control the result.

## **some removal of logic in GSPs, moving to controller/services (much more needed)**

Groovy Servlet Pages (.gsp files) cleaned up to avoid interpreted code critical to functionality

Standard indenting of HTML within GSP pages.

## **updated sql queries to include schema+table instead of adding many db synonyms**

Especially in Oracle code, tranSMART 16.3 defined synonyms to allow reference to tables without the schema.

Explicit schema references are cleaner.

They also make it possible to derive the permissions needed for functions/procedures to operate across schemas.

## **removed many unused methods, some unused classes, org.json source classes, duplicate classes (e.g. many in both transmartApp and transmart-java)**

Many unused methods were retained because it is difficult to be certain they will never be invoked.

The level of testing undergone by transmart 19.0 makes this an ideal time to remove these methods and check they were indeed redundant.

However, some removed methods in the i2b2-transmart code were unused on Oracle but were required when running on postgres and have been reinstated. Examples include handling large objects as strings.

## **deleted deprecated AccessLog class and changed to use AccessLogEntry and AccessLogService**

Simplifying the access logging code

## **added simple Spring Security role hierarchy**

This requires further testing to make sure the required tranSMART functionality is supported.

## **deleted many lib directory jars and replaced with BuildConfig dependencies**

These jar files are removed from the code repository. They are downloaded through code dependencies in BuildConfig.groovy or pom.xml.

These jar files remain in the repository. They may be removed later.

## **where possible annotated classes and methods with @CompileStatic for performance**

Many classes and methods now have @CompileStatic. Testing found very few instances where the annotation had to be removed.

## **StringBuffer -> StringBuilder**

StringBuilder is used to create a string and to append to it using '<<' syntax

## **split transmart-extensions into three projects, split out transmart-core-db-tests**

See the organization of the new single transmart repository

## **moved filters classes from grails-app/conf to grails-app/filters**

Filters are now in grails-app/filters/org/transmart...

## **some SQL injection fixes**

All SQL statements in the code need careful testing to check they work for both Oracle and Postgres

## **some conversion of simple Java classes to CompileStatic Groovy**

Testing is needed to ensure that code functions as in earlier releases.

## **Oracle support**

Oracle is fully supported using the same release (12.1 or 12.2) as previous versions of tranSMART.

## **Postgres support**

This release has been tested on Postgres up to 9.6, and on Postgres 10.

To date no attempt has been made to take advantage of the new partitioning features in more recent Postgres versions. We continue to monitor these and will consider supporting them in a future release. It is likely that legacy support for the current Postgres schema will be continued in tranSMART.

## SQLserver support (potential)

TranSMART does not support SQL server.

Upgrading the schemas to include SQLserver versions of the tables and stored procedures is relatively straightforward.

Upgrading the source code to include support for a third database would require significant work, but would also test and clean many sections of code with obvious benefits to the quality and robustness of tranSMART.

## Ubuntu 18 support

Changes have been made to support installation on Ubuntu 18.04, Ubuntu 16.04 and Ubuntu 14.04.

Automated install scripts have versions for each version with only limited divergence. For example, Ubuntu 18.04 uses tomcat 8.

Targets for Ubuntu installation targets in transmart-data are updated as appropriate (for example, a different PHP version is available in Ubuntu 16).

## ETL with transmart-data load targets

### Kettle 8 support

Releases up to 16.3 were tested only with Kettle 4.2.

tranSMART now supports Kettle versions up to pdi-ce-8.2.0.0

Only minor updates to Kettle scripts were needed to satisfy and additional validation.

### New ETL targets

New targets are in preparation

#### Browse Tab Program Metadata

A set of utility scripts are mad einto a load target to create a new Program under the Browse tab.

The disease and therapeutic area fields are validated on loading.

#### Browse Tab Study Metadata

A set of utility scripts to load study metadata for the Browse tab can now be invoked as load\_browse targets.

The input files can be created for studies in the existing curated data library.

The input data includes the text from GEO (reduced to 2000 characters), disease and therapeutic area, number of patients, citation details, study type and objectives, etc.

The program must be loaded before the study

#### Browse Tab Assay Metadata

A load target can add Assay metadata into the Browse tab.

The platform information is validated against the database ontologies before loading.

#### Sample Explorer Tab

A set of utility scripts are in preparation to load sample data into the Sample Explorer tab.

The input files can be made available for studies in the existing curated data library.

In GEO samples have limited informations but at least includes sample ID and organism.

## Coding standards

Cleanup of SQL source code

- Standard indentation of lines and within SQL statements
- Consistent use of upper and lower case

- Commas in lists of columns/values positioned to ensure the correct row is indicated in SQL error messages
- Changed 'select ... into rtncd' to 'perform' where the rtncd value was ignored.
- Datatypes cleaned up e.g. int v. bigint
- Inputs and output matched to closest equivalents in Kettle
- Added 'explain' blocks for postgres to improve performance of slowest steps

## **RNAseq ETL performance**

A missing condition in a SQL statement caused RNAseq gene expression to load extremely slowly, and to consume very large memory and tmp space resources. No other datatypes were affected.

## **cz\_job\_audit message position**

Many messages report "loading" with a row count for the previous step.

All such messages should report the end of the step with its row count and descriptions.

## **Function/stored procedure error checks**

Several stored procedures reported errors, and returned an error status, but this was ignored by the calling procedure.

Release 19.0 checks the return status for all calls that have a return value.

## **Leading zeroes in Kettle job output**

The job\_id and log\_base values were reported with a large number of leading zeroes.

The datatypes have been changed and specific formats introduced to report only the integer value.

## **Validation of platform**

All platform annotation files should include only one platform.

## **RNAseq multiple procedures**

tranSMART has two sets of RNAseq ETL procedures. One is for gene-based expression counts, the other is for expression mapped to chromosome position.

The names could be made clearer. We may rename one or both before the final release

Multiple values should be reported. They are usually caused by one or more bad input records.