Source Code Delivery Instructions

2-7 - 03 Mar 2022

The informational material contained in this section is provided as a courtesy for use by CAST's clients. Licensing details are documented on this page: https://www.gnu.org/licenses/lgpl-3.0.en.html

The official CAST AIP technologies documentation is published here

Introduction

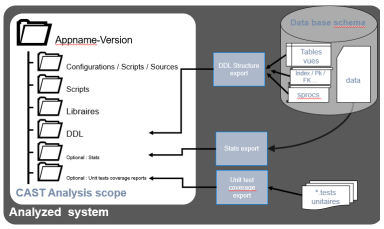
Recommendation :

The delivery structure must correspond to your project structure and must be reproducible in order to allow for consistency during subsequent analysis.

Ask your development team to get a standard export from the code repository (example: git clone).

CAST AIP analysis platform performs a static analysis of the source code and builds a model of the application to verify its structural quality.

The delivery should consist of one zip file, with a structure similar to the one in this illustration.



To perform the analysis, you must deliver all the elements necessary for compilation for the code generated upstream for the application.

- Files written by developers or generated by the IDE (project file, source, ...),
- The code files that are generated during compilation (eg: generated services), in order to have all the components of the system in production.
- The generated code must be structured at the time of delivery and the identification methods defined (naming rule, directory, etc.)
- Third-party libraries and referencing files for these libraries (pom.xml, ...)
- **NOTE:** CAST will not retrieve third-party libraries in public/private repositories (maven, npm, pip, nuggets ...) and the code will not be compiled by CAST.

It is important to exclude files that should not be scanned, for example:

- Test code
- HTML / JS files generated by documentation generation or analysis tools
- Large data files included in your deployment, but not required for the structure of the application (example: video included in a CRM, images)

For SQL databases only one DDL script per schema must be provided

- Incremental database definitions are not supported.
- Each file must start with the instruction to create the schema with a name specific to the schema (identifiable by the developers)
- The script must contain all the elements that make up the schema (Tables, fields, integrity constraints, indexes, stored procedures, etc.) but no data (no INSERT or UPDATE)
- (Optional) you may be asked for the size statistics of the tables in production in order to modulate the recommendations according to the volumes of data stored
- (Optional) you may be asked to deliver unit test code coverage reports

Details per technology

The section below details the files expected for an analysis. Please contact the project manager in case of doubt whether a type of file is suitable or not to perform static analysis.

- C C++
- Adobe (Flex)
- Fortran
- Java JEE
- Message Queues
- Mobile
- IBM Mainframe zOS (Cobol, JCL, CICS)
- EGL
- Microsoft .NET
- Microsoft (not .NET)
- Oracle Enterprise Solutions
- PHP
- IBM PL/1
- Python
- IBM System i
- Cobol (not IBM)
- SAP
- Scripting Languages

- SQL - Database
- Misc. Enterprise Solutions
- Web
- Node.js

Note: For JavaScript and HTML5, please send the files in a "normal" format rather than a minified or bundled format (minified files i.e. ***.min.js files** or bundled files containing one or a few very long lines javascript files being **automatically excluded from the analysis)**

C C++

Requirement	CAST Comment
All source files of the application, including project files. Typical extensions are .c, .cpp, .cxx, .pc… and for include : .h, .hh, .hpp, .tpp, .inl, .inc,…	• Avoid delivering .lib, .a, .obj, .so, .dll, .exe, … (libraries and executable files).
The system headers of the Operating System and compiler with which the source code is compiled.	
The external (i.e third-party) libraries headers	• You must use the header files that are consistent with the chosen Operating System (i.e the ones that are installed on the chosen Operating System to build the application).
All makefiles	• All files used by the build system (autoconf, cmake, makefiles, vcxproj, xcodeproj…)
Build log files	• Only in the case of non-Microsoft compilers. The compiler log files will be easier to read to detect the macro used for compilation, as well as the include paths.
Include path	
Set of hard-coded macros defined by the compiler	• Possibly depends on compiler options. You should provide the set corresponding to the options used to build the executables and libraries.
Macros passed as compiler or build options	• For instance, Macros can be passed on the command line as parameters (-DMACRO=value)
Executables and libraries organisation	• Provide information on the source code used to build the executables and the libraries, as well as their dependencies.

Database access API	• Which databases the C/C++ code relies on as well as what releases or versions of the aforementioned databases are being used.
	• By what means the code interacts with the database:
	  • Through embedded SQL (like PRO*C);
	  • Using the Database standard API (like OCI or OCCI for Oracle, ODBC for SQLServer, CT-LIB for Sybase…).

Dynamic code	• Is there any dynamic code (dynamic library? • What are the API calls for loading the dynamic libraries and execute the dynamic code?

Adobe (Flex)

	Requirement	Comment
Flex	The source code must be in a file/folder structure that is able to be complied. Allowable file extensions are: *.as , *.mxml	Source code often refers to include files or other layer's executable with the relative path, hence the relative paths must be respected in the delivery folder structure.

Fortran

Requirement	Comment
The source code must be in a file/folder structure that is able to be complied.	• Source code often refers to include files or other layer's executable with a relative path, hence the relative paths must be respected in the delivery folder structure.
The allowed extensions for FORTRAN files are: *.for *.FOR *.inc *.INC *.pfo *.PFO *.f *.F *.f03 *.F03 *.f90 *.F90 *.f77 *.F77 *.h	

Java JEE

Requirement	Comment
All the **source files of the application (all .java files)**, including project files and configuration files for example • .XML files used by frameworks • .properties, .json, yml files • EJB configuration files and all their deployment mapping files • All associated Deployment Mapping Files (required to link with the database). • web.xml for each web application • All .jsp / .jcx / .jspf / .jhtml and similar types of files in the application • All xsl files • All Tags libs (.tld) • All XML configuration files (struts-config.xml, ...) and mapping files. • All DTD files associated to the XML files • All HTML files • All JavaScript files • All css and (image files can be removed) • for web services, WSDL/WSDD/mapping files • TLD documentation	All the source code must be encoded in ASCII or in a supported Unicode encoding (UTF-8, …) Supported encoding/character sets are listed Supported Character Sets. When collecting application source code, ensure that you will have the .java files and not .class or .jar files for source code within the scope of the analysis. .class and .jar files are decompiled only to extract the signature.
Build configuration files : Maven and/or Eclipse project files : *.pom, *.project, *.classpath Gradle, for information purpose Ant, Ivy scripts should be delivered but will not be analyzed automatically	Including complete hierarchy of ALL .pom and parent .pom files. These files will allow the automated discovery and analysis configuration.
All libraries (.JAR) required to build the application including JDK, JEE, related libraries e.g. application server, 3rd party, framework, presentation, persistence, etc.. In the case of a maven project: Provide a **copy of the local maven repository**	Custom JAR files, included e.g. custom J2EE.JAR, used for Presentation and Persistence should be delivered with the source code. Failure to do so will prevent transaction identification and all depending metrics.

.JAR files .WAR or .EAR should not be delivered	Required to reproduce the real production structure with the sources.
If some SQL files remain in the source delivery, you must explain how to deal with them.	Normally a database must be extracted as DDL files and delivery as SQL files in a specific folder/zip
Files that contain data are useless (*.dat, *.doc, *.ppt, *.xls, *.xml that contains data only)	These files are big and not required, not analyzed. So there is no need to give them.

Copy of the local maven repository is not mandatory, only Jars files in a repository that are used by the project are mandatory.

Exclude .git, .svn folders

Message Queues

- ActiveMQ (Java, Python)
- IBM MQ (Java, Python)
- RabbitMQ (Java, Python)
- Spring JMS
- Spring APMQ

	Requirement	Comment
ActiveMQ (Java, Python)	All configuration files	In addition to provided application files
IBM MQ (Java, Python)	All configuration files	
RabbitMQ (Java, Python)	All configuration files	
Spring JMS	All configuration files	
Spring APMQ	All configuration files	

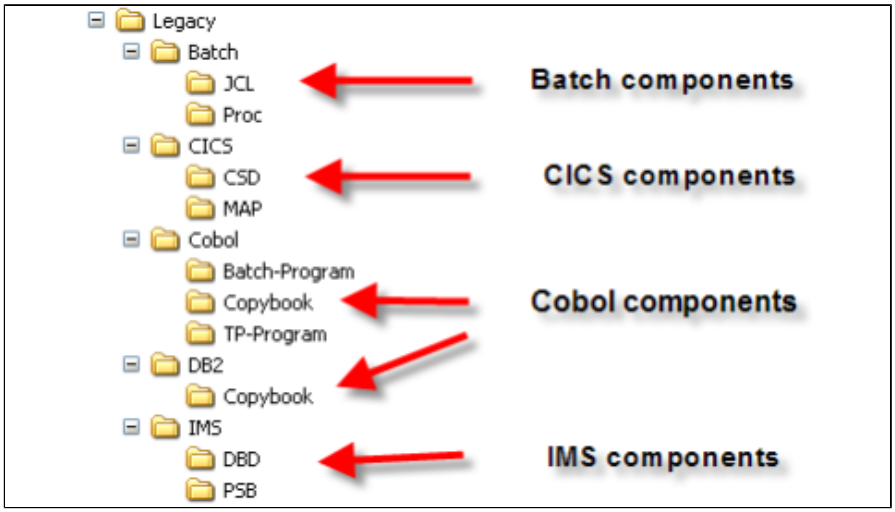
Mobile

- Android
- iOS/objective C
- iOS/Swift

	Requirement	Comment
Android	All application files + configuration files	
iOS/objective C	All application files + configuration files	
iOS/Swift	All application files + configuration files	
ionic	Source code before compilation Files extensions are *.ts, *.html, *.css or *.scss	Source code written by developers is mandatory
React Native	Source code before compilation Files extensions is *.js only	Source code written by developers is mandatory If there is another specific file extension, it must be clarified by the customer

IBM Mainframe zOS (Cobol, JCL, CICS)

	Requirements	**Comments**
Cobol, JCL, CICS - Option 1	Export of PDS (when the programs are members of a Partitioned Data Set)	• Use IEBPTPCH to print and punch the PDS (1 PDS per type of source code) before transfer to PC. (Additional detail can be found in the Partitioned Data Set (PDS) section on this page: Mainframe - application qualification specifics#applicationqualificationspecifics-PartitionedDataSet)
	All the source code must be in ASCII format	• If files are not in ASCII format or CR/LF are not readable, then the Mainframe Analyzer will not able to analyze them.

	TRANSACTION (. CSDs and .BMSs)	• .CSD: This resource definition can be delivered via a copy (into an ASCII flat file) of the script used to define the CICS environment or by using a JCL in order to extract this information from CICS. See an example of JCL code to extract the CSD and a sample JCL using the IBM utility DFHCSDUP at this link On-line processing . • .BMS: description of the mapset
	All the source code must be in ASCII format	• If files are not in ASCII format or CR/LF are not readable, then the Mainframe Analyzer will not able to analyze them.
Cobol, JCL, CICS - Option 2	Each application must be delivered in a single folder. 1 file per object type.	• 1 file per program, 1 file per copybook, 1 file per JCL job, etc in the folder structure as shown in the image below 
	All the source code must be in ASCII format	• If files are not in ASCII format or CR/LF are not readable, then the Mainframe Analyzer will not able to analyze them.
	Each application folder must contain sub-folders split by technology as below:	
	• COBOL	• Cobol programs and copybooks
	• BATCH	• JCL jobs, JCL includes, JCL procedures
	• TRANSACTION	• .CSD: This resource definition can be delivered via a copy (into an ASCII flat file) of the script used to define the CICS environment or by using a JCL in order to extract this information from CICS. See an example of JCL code to extract the CSD and a sample JCL using the IBM utility DFHCSDUP at this link On-line processing. • .BMS: description of the mapset
	• IMS	• IMS database description (DBD) as a text file. • PSB files including all the PCB of the application
	Allowable File Extensions :	
	• COBOL programs	*.cbl, *.cob and *.cobol

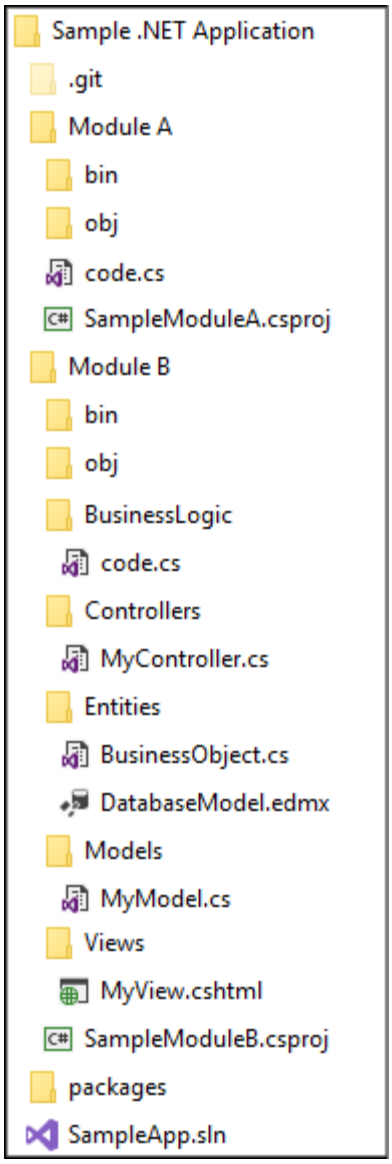
• COPYBOOK files	*.cpy and *.copy
• IMS files	*.psb and *.dbd
• CICS files	*.csd and *.bms
• JCL Jobs	*.jcl
• JCL Procedures	*.prc and *.proc
• JCL Includes	*.inc

EGL

Requirement	Comments
All EGL programs files must be under one folder structure. Permitted files extension are : *.egl *.egldd *.eglbld *.project	The sub-directory structure can reflect the application organization.

Microsoft .NET

	Requirement	Comment

All artefacts required to build the system must be delivered, including:
• Project files: *.sl, *.csproj, *.vbproj
• .NET code files: *.cs, *.vb
• GUI files:
  • for ASP.NET Applications: *.aspx, *.ascx
  • for ASP.NET MVC Applications: *.cshtml, *.vbhtml
  • for WPF Applications: *.xaml
• Additional assets:
  • for instance *.htm, *.html, *.css, *.js
• Configuration files:
  • *.config, *.properties
  • for Entity Framework: *.edmx (if applicable)
• 3rd party libraries: assemblies are used by the application which are not part of .NET Framework
  • *.dll
  • for COM/ActiveX components: *.ocx
	GENERAL GUIDELINES
	• Usually, source code is versioned using a CMS (tools like SVN or Git)
	• The best way to deliver source code is to use the CMS and check out the source code of the version that has been selected for analysis
	  • Make sure to NOT include the ".git" or ".svn" folders
	• Regarding 3rd party libraries :
	  • If dependencies are managed via Nuget, they should be located in one or more "packages" folders zip those folders
	  • Another way to provide assemblies is to provide a zip with the compiled version of the application (including the application's assemblies AND the external assemblies)
	
	EXAMPLE
	
	Given the organization pictured on the right, the full content of the "Sample .NET Application" folder must be provided, except:
	• ".git" folder: related to CMS used by the development team (here Git).
	• "obj" folders: they contain temporary/intermediate artefacts generated during the build process.
	
	To deliver 3rd parties assemblies, the "bin" folders and/or the "packages" folders may be included in the delivery.
	• "bin" folders contain binaries (debug and/or release assemblies) that are created at build time. They should also contain third-party assemblies.
	• "packages" folders are created by the Nuget package manager and contain assemblies and artefacts that are published in a Nuget repository.
		

Microsoft (not .NET)

- ASP
- VB

	Requirement	Comment
ASP	All the source code must encoded in ASCII or a supported Unicode encoding e.g. UTF-8	• Supported encoding/character sets are listed at this link Supported Character Sets
	All .asp / .inc ASP pages of the web application	• Including included files (.inc)
	All .asa / .htm / .html / .htc / .css / .vbs files	
	All .htm / .html / .js / .vbs files	• Client code
	All .gif / . Bmp / .png / .jpg / .jpeg files	• Images used by the application

	COM objects used by the above pages : .ocx, .dll, .exe	• Components for which you don't have the source, including third-party components
VB	All the source code must be encoded in ASCII or a supported Unicode encoding e.g. UTF-8	Supported encoding/character sets are listed Supported Character Sets .
	Projects	• .vbp
	Groups	• .vbg (if any)
	All Source code referenced by the projects: .bas, .frm , .cls, .ctl,…	• Including third-party components
	COM objects used by the application: .ocx, .dll, .exe	• Components for which you don't have the source, including third-party components

Oracle Enterprise Solutions

- Oracle Forms/Reports
- PeopleSoft
- Siebel

	Requirement	Comment
Oracle Forms /Reports	DMT's Oracle Forms and Reports option must be used extraction. Oracle Forms .FMB .OLB .MMB .PLL Oracle Reports .REX	
PeopleSoft	To download the PeopleSoft extractor go to CAST Extend : https://extend.castsoftware.com/V2/ The extractor is composed of the following elements : • CASTDBGUI.jar • CASTPeopleSoft_db2.config • CASTPeopleSoft_orcl.config • PeopleSoft-Extract-CLI.bat : all the row with <PARAM> should be fulfill with right parameter (path, DB name,..) The above elements should be delivered to CAST : • Extraction of Vanilla environment • Extraction of Project environment • DDL of the Database	The specific code into PeopleSoft it is not detectable, it is mixed with the PeopleSoft code. To be able to analyze the specific code, CAST will do a comparison between the original PeopleSoft code (called Vanilla), and the PeopleSoft with the specific code (called Project). The two environments should be exactly in the same release, and extractions should be done with the CAST PeopleSoft extractor for Vanilla and Project.
Siebel	The CAST SIEBEL extractor must be used for source code extraction.	

PHP

Requirement	Comment

All the files .html .htm .phtml .php .php4 .php5 . php6 .inc .js (JavaScript) and .css of the application.	• The analysis of XML and XSL files contained in the PHP application is not supported. Those files can be delivered. • The analysis of HTML and JavaScript source code is managed by the JEE or the New .NET analyzer, to be configured in addition to the generated UA for PHP job.
All .gif .bmp .png .jpg .jpeg .tiff files	• Images used by the application

For Php specify the framework used (symfony, drupal, joomla etc....)

IBM PL/1

Requirement	Comment
Each file must contain only a single artefact or component	• A file can only contain a PL/1 program or a single Include file
DB2 table definitions must be exported to include files (PLC extension).	• The 'EXEC SQL INCLUDE' referring to include files that are generated by DB2 for PL1 are targeted.
The type of source code must be identifiable using the extension of the file.	
All PLI program files must be under one folder structure.	• The sub-directory structure can reflect the application organization. • All include files must be under a PLC directory. • All table definition files must be under a TLC directory.
PL1 programs must have PLI extensions.	
Include and table definition files must have PLC extensions	

Python

	Requirement	Comment
Python Core	The source code for **Python** need to be delivered with the following file extensions : *.py	
Python dependencies	The file containing the list of python dependencies (requirements.txt)	

IBM System i

- RPG
- Cobol AS400

Note: some artefacts, like Physical Files (*.PF), Logical Files (*.LF) are used by COBOL for AS400, but are parsed by RPG extension. Some Links may be added from COBOL to

	Requirement	Comment

RPG	Each application must be delivered in a single folder
	Each application folder must contain sub-folders split by technology as below :
	
	• RPG programs from QRPGSRC library
	  • source member with file extensions of *.RPG (RPG-III), *.RPGLE, *.SQLRPGLE
	• COPY RPG from QCPYSRC library (for rpg and cl programs)
	  • source member with file extensions of *.CPYRPG, *.CPYCLP
	• CL programs from QCLPSRC library
	  • source member with file extensions of *.CLLE, *.CLP
	• Display Files from QDSPSRC library
	  • source member with file extensions of *.DSPF
	• Printer files from QRPTSRC library
	  • source member with file extensions of *.PRTF
	• Logical File (LF) and Physical Files from QDDSSRC
	  • source member with file extensions of *.LF, *.PF
		For the complete list of file extensions, see RPG extension documentation
		
		Extraction source code from AS400 libraries.
		
		The QCPYSRC library can contain copies for RPG, CL and COBOL programs

Cobol AS400	Each application must be delivered in a single folder
	Each application folder must contain sub-folders split by technology as below :
	
	• COBOL programs from QCBLSRC library
	  • source member with file extensions of *.CBL, *.SQLCBL, *.SQLCBLPCML, *.SQLCBLLET, *.SQLCBLLEC
	• COPY COBOL from QCPYSRC library
	  • source member with file extensions of *.CPYCBL
		Extraction source code from AS400 libraries.
		
		The QCPYSRC library can contain copies for RPG, CL and COBOL programs.

datastage	- dsx export of the jobs - dsx exports of the stage types	

Cobol (not IBM)

- Microfocus Cobol
- GCOS Cobol
- Cobol VMS

	Requirement	Comment
Microfocus Cobol	All files used to build the system. Exact file types to be confirmed.	Please consult your CAST technical contact.
GCOS Cobol	All files used to build the system. Exact file types to be confirmed.	Please consult your CAST technical contact.
Cobol VMS	All files used to build the system. Exact file types to be confirmed.	Please consult your CAST technical contact.

SAP

- SAP ABAP
- SAP UI5 (Fiori)
- SAP Hybris
- SAP BusinessObjects - Reporting Enterprise Solutions

	Requirement	Comment
SAP ABAP	SAP ABAP extracted using CAST SAP Extractor NG	The CAST ABAP extractor extracts ABAP source code, the database structure (tables, index, keys, ...) and can also grab the table sizes. Various options for extraction exists : per package, per date, per Transport Request, ...
SAP UI5 (Fiori)	Same requirements as for web / HTML5 / jQuery technologies	
SAP Hybris	Same requirements as for Java/JEE technologies	
SAP BusinessObjects - Reporting Enterprise Solutions	DMT's Business Objects option must be used for source code extraction. DMT will require location of Business Objects Universes (.unv files) for extraction.	

Scripting Languages

- Korn shell
- Bourne shell
- C shell

	Requirement	Comment
Korn shell, Bourne shell, C shell	Each file must contain only a single artefact or component	

SQL - Database

- T-SQL (Sybase and Microsoft SQL Server)
- Informix
- MariaDB
- MySQL
- PostgreSQL
- Oracle Server PL/SQL
- SQLite
- Teradata
- IBM DB2

Regarding documentation https://doc.castsoftware.com/display/TECHNOS/SQL+-+Covered+technologies

For those technologies, one file containing a DDL script containing per database/schema creation statements are requested, including constraints (PK,FK), Indexes, Store Procedures, triggers ...

Notes :

1. Incremental schema creation is not supported, and incremental scripts should be removed from the delivered code.
2. Data creation/initialization must not be included in the script.

	Requirement	Comment
T-SQL (Sybase and Microsoft SQL Server)	DDL script creation is required. The following link will give you DDL extraction examples	• DDL schemas will be analyzed using SQL analyzer • The sqltablesize file generation script, along with pertinent details is available in Enable XXL table Quality Rules#. sqltablesizefilegenerationmethods.
	SQL Server/Sybase user used for extraction must have a user with the privileges specified in the Microsoft SQL Server/Sybase ASE sections in Required RDBMS rights for packaging a database .	
Informix	DDL script creation is required. The following link will give you DDL extraction examples	• DDL schemas will be analyzed using SQL analyzer • The sqltablesize file generation script, along with pertinent details is available in Enable XXL table Quality Rules#. sqltablesizefilegenerationmethods.
MariaDB	DDL script creation is required. The following link will give you DDL extraction examples	• DDL schemas will be analyzed using SQL analyzer • The sqltablesize file generation script, along with pertinent details is available in Enable XXL table Quality Rules#. sqltablesizefilegenerationmethods.

MySQL	DDL script creation is required. The following link will give you DDL extraction examples	• DDL schemas will be analyzed using SQL analyzer • The sqltablesize file generation script, along with pertinent details is available in Enable XXL table Quality Rules#. sqltablesizefilegenerationmethods.
PostgreSQL	DDL script creation is required. The following link will give you DDL extraction examples	• DDL schemas will be analyzed using SQL analyzer • The sqltablesize file generation script, along with pertinent details is available in Enable XXL table Quality Rules#. sqltablesizefilegenerationmethods.
Oracle Server PL /SQL	DDL script creation is required. The following link will give you DDL extraction examples	• DDL schemas will be analyzed using SQL analyzer • The sqltablesize file generation script, along with pertinent details is available in Enable XXL table Quality Rules#. sqltablesizefilegenerationmethods.
	Oracle user used for extraction must either be SYSTEM or a dedicated extraction user with the privileges specified in the Oracle Server /Extraction user section in Required RDBMS rights for packaging a database .	
SQLite	DDL script creation is required. The following link will give you DDL extraction examples	• DDL schemas will be analyzed using SQL analyzer • The sqltablesize file generation script, along with pertinent details is available in Enable XXL table Quality Rules#. sqltablesizefilegenerationmethods.
Teradata	DDL script creation is required. The following link will give you DDL extraction examples	• Attached sample script which is based on BTEQ querytool - Script • Needs the Teradata DB host name, user name, password, DB names used by the application
IBM DB2	DDL script creation is required. The following link will give you DDL extraction examples	• DDL schemas will be analyzed using SQL analyzer • The sqltablesize file generation script, along with pertinent details is available in Enable XXL table Quality Rules#. sqltablesizefilegenerationmethods.

NoSQL

- MongoDB
- Couchbase
- Marklogic
- CouchDB
- AzureCosmosDB (Java, .net)
- DynamoDB (Java, .net)
- Elasticsearch (Java, .net)
- Memcached (Java, .net)
- Redis (Java, .net)

	Requirement	Comment

MongoDB	MongoDB is accessed via the different frameworks written in the language of the application so there is no specific types files related to MongoDB. For the JEE application the access to MongoDB will be done in java files (refer to Java applications requirements) For the .Net application the access to MongoDB will be done in .net files (refer to .Net applications requirements) For the NodeJS application the access to MongoDB will be done in JavaScript files (refer to NodeJS applications requirements)	
Couchbase	Couchbase is accessed via the different frameworks written in the language of the application so there is no specific types files related to Couchbase. For the JEE application the access to Couchbase will be done in java files (refer to Java applications requirements)	
Marklogic	MarkLogic is accessed via the different frameworks written in the language of the application so there is no specific types files related to MarkLogic. For the NodeJS application the access to MarkLogic will be done in JavaScript files (refer to NodeJs applications requirements)	
CouchDB	CouchDB is accessed via the different frameworks written in the language of the application so there is no specific types files related to CouchDB. For the NodeJS application the access to CouchDB will be done in JavaScript files (refer to NodeJs applications requirements)	
AzureCosmosDB		
DynamoDB		
Elasticsearch		
Memcached		
Redis		

Misc. Enterprise Solutions

- TIBCO
- SAP Power Builder
- Oracle BPEL

	Requirement	Comment
TIBCO	Project configuration files : • *.aeschema • *.sharedjdbc • *.javaschema • *.sharedhttp • *.sharedjmscon • *.process • *.wsdl • *.sharedjmsapp All Java source files of the application All properties and XML files Folder structure must be one sub folder per project	Typically all these files are under the same root folder.
SAP Power Builder	All the source code must encoded in ASCII or a supported Unicode encoding e.g. UTF-8	• Supported encoding/character sets are listed Supported Character Sets.
	PowerBuilder Libraries (.pbl)	• All PowerBuilder libraries from the application (and the framework if any)

	If PowerBuilder version < 8 : file pb.ini If PowerBuilder version >= 8: file xxx.pbw and xxx.pbt	• pb.ini will contain the starting PBL and application's name • .pbw is the PowerBuilder Workspace • .pbt is the PowerBuilder Target
Oracle BPEL	Source files • *.bpel • *.wsdl • *.java	• .bpel file contains the steps/activities involved in the business process • .wsdl file contains the definition of the web-services orchestrated by the BPEL file • .java contains the implementation of the web-services defined in the WSDL file

Web

- Angular
- Angular.JS
- JavaScript
- jQuery
- ReactJS
- TypeScript
- VBScript

Note: For JavaScript and HTML5, please send the files in a "normal" format rather than a minified or bundled format (minified files i.e. ***.min.js files** or bundled files containing one or a few very long lines JavaScript files being **automatically excluded from the analysis)**

	Requirement	Comment
Angular.JS	All application files + configurations files Code before compilation (else there will be generated code in the delivery). All the source files of the application, including project files and configuration files for example: • .js - code source files • .properties, .config, .json, .yml - properties and configuration files • .html files • .css, .scss files • images, icons aso files All libraries required to build the application, with the exception of external libraries which are not included in the analysis.	*.yaml, *.js, *.jsx, *.css, *.ts, *.tsx, *.json, *.html ...
Angular	Source code before compilation Files extensions are *.ts, *.html, *.css or *.scss. Source folder that contains index.html, main.ts and the app folder app.modules.ts file spec.ts files are useless. Don't deliver them.	Source code written by developers is mandatory. If there is other specific file extension, it must be clarify by customer.
JavaScript	All application files + configurations files	*.yaml, *.js, *.jsx, *.css, *.ts, *.tsx, *.json, *.html ...
jQuery	jquery file not the minified file. add jQuery version used	*.yaml, *.js
ReactJS	Source code before compilation Files extensions is *.js only *.spec.js and *.test.js and *-spec.js and *-test.js are useless. Don't delivery them	Source code written by developers is mandatory. If there is other specific file extension, it must be clarify by customer.
TypeScript	All application files + configurations files	*.yaml, *.js, *.jsx, *.css, *.ts, *.tsx, *.json, *.html ...
VBScript	All application files + configurations files	*.vbs

Node.js

Requirement	Comment

All the source files of the application, including project files and configuration files for example: • .js, .jsx, .ts, .tsx - code source files • .pug, .jade, .vue aso - files used by frameworks • .properties, .config, .json, .yml - properties and configuration files • .html files • .css, .scss, .sass files • images, icons aso files	All the source code must be encoded in ASCII or in a supported Unicode encoding (UTF-8, …)
Project files: package.json, package-lock.json	

Note: do not provide a copy of the **node_modules/** directory

To go further

This page details the source code file types required per technology to be delivered. This document is used by the CAST Administrator during Qualification and Source Code Delivery phases as detailed at this link Prepare and deliver the source code, and also for the Analysis Configuration phase as detailed at this link eBlue Print - Analysis Configuration and Execution