<div align="center">**OVERVIEW**</div>

The Razor tool suite includes the following primary components:
- **Versions**. A file version control system.
- **Threads**. A build or release management tool.
- **Issues**. A problem tracking tool.

The Razor tools may be used collectively as a tool suite or independently. The tools share a common database. The database is composed of one or more issues groups, one or more file-control groups, and a project group. File and thread groups are linked, with a single project group existing for all file/thread pairs in the database. Issues can be related to version control and thread activity, or can be used in a standalone fashion. Therefore, it is envisioned that one approach for interfacing the activity stored by Razor with third party tools would be to utilize the issue control groups for "defect" tracking; version control groups for "versioning" and threads control groups for "release" or "baseline".

Based upon this approach, the proposed API provides four categories of commands:

1. Utility commands related to collection of activity for Versions and Threads.
2. Versions and Threads activity collection commands
3. Utility commands related to collection of activity for Issues
4. Issues activity collection commands.

This would mean that for a client-sever architecture the integration would need to take place on the same host or server where the Razor database resides. The client and server would exchange text commands and results over standard socket interface.

UTILITY COMMANDS FOR COLLECTION OF VERSIONS AND THREADS ACTIVITY

`rz_isRUD –rud <RUDirPath>`
RUDPath – The Razor universe directory (RUD) path locating a particular Razor database.
Purpose: Function validates path to specified database.
Returns: On success, the valid path is returned. Otherwise, an error message is returned.

`rz_getFileControlGroups –rud <RUDPath>`
RUDPath – The Razor universe directory (RUD) path locating a particular Razor database.
Purpose: Obtains list of all file control groups associated with the Razor database.
Returns: Comma-delimited list of Razor file control groups for the specified Razor database.

`rz_isFileControlGroup –rud <RUDPath>  –g <GroupName>`
RUDPath – The Razor universe directory (RUD) path locating a particular Razor database.
GroupName – Name of Razor file control group to be validated.
Purpose: Provides a means of validating a given string as the name of a current Razor file control group.
Returns: True/False

`rz_getAttributes –rud <RUDPath>  –g <GroupName>`
RUDPath – The Razor universe directory (RUD) path locating a particular Razor database.
GroupName – Name of Razor file control group.
Purpose: Razor attributes can be customized down to the group level. Function serves as an accessor obtaining the name and type of each attribute being used for the specified database.

Returns: A comma-delimited list of attribute name/type-value pairs. E.g. Blessing/STATE, Filetype/ONE_OF_MANY.

`rz_getAttributeValues –rud <RUDPath> –g <GroupName> -a <AttributeName> [-echo] [-verbose]`
RUDPath – The Razor universe directory (RUD) path locating a particular Razor database.
GroupName – Name of Razor file control group.
AttributeName - Name of valid attribute for specified group.
Purpose: Obtains list of permissible attribute values for an attribute of type CHECK_BOX, ONE_OF_MANY, ONE_OF_MANY_STACK, X_OF_MANY, STATE, ONE_OF_MANY_LIST, X_OF_MANY_LIST.
Returns: On success returns a comma-delimited list of attribute values.

`rz_isValidAttributeName –rud <RUDPath> –g <GroupName> -a <AttributeName> [-echo] [-verbose]`
RUDPath – The Razor universe directory (RUD) path locating a particular Razor database.
GroupName – Name of Razor file control group.
AttributeName - Name of attribute to be validated.
Purpose: Provides a means of validating given attribute for specified Razor group.
Returns: TBD

`rz_isValidAttributeValue –rud <RUDPath> –g <GroupName> -a <AttributeName>:<AttributeValue> [-echo] [-verbose]`
RUDPath – The Razor universe directory (RUD) path locating a particular Razor database.
GroupName – Name of Razor file control group.
AttributeName: AttributeValue - Name of attribute name-value pair to be validated.
Purpose: Provides a means of validating given attribute name-value pair for specified Razor group.
Returns: TBD

VERSIONS AND THREADS ACTIVITY COLLECTION COMMANDS

`rz_getTotalFiles –rud` *`<RUDPath>`* `[–g` *`<GroupName>`*`] [-ts` *`<SinceTimestamp>`*`] [-tu` *`<UntilTimestamp>`*`] [-echo] [-verbose]`
`RUDPath` – The Razor universe directory (RUD) path locating a particular Razor database.
`GroupName` – Optional Razor file control group for the query.
`SinceTimestamp, UntilTimestamp` – Optional timestamp(s) used to window the query.
Timestamps are to be formatted as follows: yy/mm/dd,hh:mm:ss.
Purpose: Obtains total number of files
Returns: Total number of files.

`rz_getTotalSubfolders –rud` *`<RUDPath>`* `[–p` *`<AuxPath>`*`] [–g` *`<GroupName>`*`] [-echo] [-verbose]`
`RUDPath` – The Razor universe directory (RUD) path locating a particular Razor database.
`AuxPath` – Optional auxiliary path from which subfolders are to be located. Path should be specified relative to the RUDPath. Cannot utilize both the "-p" and "-g" options within the same query.
`GroupName` – Optional Razor file control group for the query. Cannot utilize both the "-p" and "-g" options within the same query.
Purpose: Identifies subfolders within Razor database, directory, and/or group.
Returns: Total number of subfolders

`rz_getFilesByAttrVal –rud` *`<RUDPath>`* `–g` *`<GroupName>`* `-a` *`<AttributeName>`*`:`*`<AttributeValue>`* `[-ts` *`<SinceTimestamp>`*`] [-tu` *`<UntilTimestamp>`*`] [-echo] [-verbose]`
`RUDPath` – The Razor universe directory (RUD) path locating a particular Razor database.
`GroupName` – Razor file control group for the query.
`AttributeName: AttributeValue` - Name of attribute name-value pair for the query.
`SinceTimestamp, UntilTimestamp` – Optional timestamp(s) used to window the query.
Timestamps are to be formatted as follows: yy/mm/dd,hh:mm:ss.
Purpose: Provides a count for the number of files under source code control with a matching attribute value.
Returns: <mark>TBD</mark>

`rz_getAddedFiles –rud` *`<RUDPath>`* `[–g` *`<GroupName>`*`] [-a` *`<AttributeName>`*`:`*`<AttributeValue>`*`] -ts` *`<SinceTimestamp>`* `[-tu` *`<UntilTimestamp>`*`] [-echo] [-verbose]`
`RUDPath` – The Razor universe directory (RUD) path locating a particular Razor database.
`GroupName` – Optional Razor file control group for the query.
`AttributeName: AttributeValue` - Optional attribute name-value pair for the query.
`SinceTimestamp, UntilTimestamp` – Timestamp(s) used to window the query. Timestamps are to be formatted as follows: yy/mm/dd,hh:mm:ss.
Purpose: Provides a count for the number of files added to source code control within the specified period.

Returns: <mark>TBD</mark>

`rz_getChangedFiles –rud` *`<RUDPath>`* `[–g` *`<GroupName>`*`] [-a` *`<AttributeName>`*`:`*`<AttributeValue>`*`] -ts` *`<SinceTimestamp>`* `[-tu` *`<UntilTimestamp>`*`] [-echo] [-verbose]`

`RUDPath` – The Razor universe directory (RUD) path locating a particular Razor database.
`GroupName` – Optional Razor file control group for the query.
`AttributeName: AttributeValue` - Optional attribute name-value pair for the query.
`SinceTimestamp`, *`UntilTimestamp`* – Timestamp(s) used to window the query. Timestamps are to be formatted as follows: yy/mm/dd,hh:mm:ss.
Purpose: Provides a count for the number of files under source code control that were changed within the specified period.
Returns: <mark>TBD</mark>

`rz_getUnchangedFiles –rud` *`<RUDPath>`* `[–g` *`<GroupName>`*`] [-a` *`<AttributeName>`*`:`*`<AttributeValue>`*`] -ts` *`<SinceTimestamp>`* `[-tu` *`<UntilTimestamp>`*`] [-echo] [-verbose]`

`RUDPath` – The Razor universe directory (RUD) path locating a particular Razor database.
`GroupName` – Optional Razor file control group for the query.
`AttributeName: AttributeValue` - Optional attribute name-value pair for the query.
`SinceTimestamp`, *`UntilTimestamp`* – Timestamp(s) used to window the query. Timestamps are to be formatted as follows: yy/mm/dd,hh:mm:ss.
Purpose: Provides a count for the number of files under source code control that were not changed within the specified period.
Returns: <mark>TBD</mark>

`rz_getFileSummary –rud` *`<RUDPath>`* `[–g` *`<GroupName>`*`] [-an` *`<AttrNameList>`*`] -ts` *`<SinceTimestamp>`* `[-tu` *`<UntilTimestamp>`*`] [-echo]`

`RUDPath` – The Razor universe directory (RUD) path locating a particular Razor database.
`GroupName` – Optional Razor file control group for the query.
`AttrNameList` - Optional list of additional attributes to be included in the file details.
`SinceTimestamp`, *`UntilTimestamp`* – Timestamp(s) used to window the query. Timestamps are to be formatted as follows: yy/mm/dd,hh:mm:ss.
Purpose: Provides a detailed file summary. Minimally, each summary includes: file name, number of revisions over specified time period, and current version ID.
Returns: <mark>TBD</mark>

`rz_getChangeSummary –rud` *`<RUDPath>`* `[–g` *`<GroupName>`*`] [-an` *`<AttrNameList>`*`] -ts` *`<SinceTimestamp>`* `[-tu` *`<UntilTimestamp>`*`] [-echo]`

`RUDPath` – The Razor universe directory (RUD) path locating a particular Razor database.
`GroupName` – Optional Razor file control group for the query.
`AttrNameList` - Optional list of additional attributes to be included in the file details.
`SinceTimestamp`, *`UntilTimestamp`* – Timestamp(s) used to window the query. Timestamps are to be formatted as follows: yy/mm/dd,hh:mm:ss.
Purpose: Provides a detailed summary for files changed over the specified time period. Minimally, each summary includes: file name, number of revisions over specified time period, user-name and comments for each revision.

Returns: <mark>TBD</mark>

rz_getUserActivity –rud <RUDPath> [-u <UserList>] [–g <GroupName>] [-an <AttrNameList>] -ts <SinceTimestamp> [-tu <UntilTimestamp>] [-echo]
RUDPath – The Razor universe directory (RUD) path locating a particular Razor database.
UserList – Optional (comma-delimited) list of users for which activity is desired.
GroupName – Optional Razor file control group for the query.
AttrNameList - Optional list of additional attributes to be included in the file details.
SinceTimestamp, **UntilTimestamp** – Timestamp(s) used to window the query. Timestamps are to be formatted as follows: yy/mm/dd,hh:mm:ss.
Purpose: Summarizes the changes made by each user over the specified time period. Minimally, summary includes user-name and number of changes. User summary may be expanded to include: files introduced/modified, <mark>TBD</mark>.
Returns: <mark>TBD</mark>

UTILITY COMMANDS FOR COLLECTION OF ISSUES ACTIVITY

<mark>TBD</mark>

CM METRICS COLLECTION

<mark>TBD</mark>

SAMPLE USAGE

Assume that the Razor database is configured as follows:
- Single issue control group (ICG1)
- Two file control groups (FCG1, FCG2)
- The FileType attribute has been defined for all file control groups. The permissible attribute values are: C++, Java, Docs
- The Blessing (i.e. state level) attribute has been defined for all file control groups. The permissible attribute values are: Active, Reviewed, Tested, Integrated.
- Razor database is located in /home/brianm/db1/RAZOR_UNIVERSE

Q. How do I obtain a list of the available file control groups for a particular DB?
A. Issuing command "`rz_getFileControlGroups –rud` **/home/brianm/db1/RAZOR_UNIVERSE**" returns "FCG1, FCG2"

Q. How do I obtain the attributes associated with the FCG2 file control group?
A. Issuing command "`rz_getAttributes –rud` **/home/brianm/db1/RAZOR_UNIVERSE –g FCG2**" returns
"FileType,Blessing".

Q. How can I determine the different status/state/blessing levels associated with a file under version control within the FCG1 group?
A. Issuing command "`rz_getAttributeValues –rud` **/home/brianm/db1/RAZOR_UNIVERSE –g FCG1 -a Blessing" returns "Active,Reviewed,Tested,Integrated"**

Q. How can I determine how many files are in the entire Razor DB in May 20001?
A. Issuing command "`rz_getTotalFiles –rud` **/home/brianm/db1/RAZOR_UNIVERSE** `-ts 01/05/01:00:00:00 -tu 01/05/31:23:59:59`" returns "7".

Q. What effect does the verbose option (-verbose) have on my queries?
A. Repeating the previous example. Issuing command "`rz_getTotalFiles –rud` **/home/brianm/db1/RAZOR_UNIVERSE** `-ts 01/05/01:00:00:00 -tu 01/05/31:23:59:59 –verbose`" returns
./DOMAIN_01/FCG1/inc/myTypes.h
./DOMAIN_01/FCG1/inc/moreTypes.h
./DOMAIN_01/FCG1/src/main.cc
./DOMAIN_01/FCG2/inc/otherTypes.h
./DOMAIN_01/FCG2/src/other.cc
./DOMAIN_01/FCG2/src/more.cc
./DOMAIN_01/FCG2/doc/readme.txt
TOTAL: 7

Q. How do I obtain the number of subfolders for the file control groups?
A. Issuing command "`rz_getTotalSubfolders –rud` **/home/brianm/db1/RAZOR_UNIVERSE**" yields: "`5`". Issuing command "`rz_getTotalSubfolders –rud` **/home/brianm/db1/RAZOR_UNIVERSE -verbose**" yields:
./DOMAIN_01/FCG1/inc
./DOMAIN_01/FCG1/src
./DOMAIN_01/FCG2/inc
./DOMAIN_01/FCG2/src
./DOMAIN_01/FCG2/doc
TOTAL: 5

Q. How do I obtain the number of header files currently in the "Reviewed" state from FCG1? Issuing command "`rz_getFilesByAttrVal –rud` **/home/brianm/db1/RAZOR_UNIVERSE –g** `FCG1 -a FileType:C++,Blessing:Reviewed`" returns "2". Issuing command "`rz_getFilesByAttrVal –rud` **/home/brianm/db1/RAZOR_UNIVERSE –g FCG1 -a** `FileType:C++,Blessing:Reviewed –verbose`" returns:
./DOMAIN_01/FCG1/inc/moreTypes.h
./DOMAIN_01/FCG1/src/main.cc
TOTAL 2