Integrating the Healthcare Enterprise

IHE Radiology
Technical Framework Supplement

Management of Radiology Report Templates (MRRT)

Trial Implementation

Date: June 11, 2013

Authors: IHE Radiology Technical Committee
Email: radiology@ihe.net

Foreword

This is a supplement to the IHE Radiology Technical Framework V11. Each supplement undergoes a process of public comment and trial implementation before being incorporated into the volumes of the Technical Frameworks.

This supplement is published on June 11, 2013 for Trial Implementation and may be available for testing at subsequent IHE Connectathons. The supplement may be amended based on the results of testing. Following successful testing it will be incorporated into the Radiology Technical Framework. Comments are invited and may be submitted at http://www.ihe.net/radiology/radiologycomments.cfm.

This supplement describes changes to the existing technical framework documents.

"Boxed" instructions like the sample below indicate to the Volume Editor how to integrate the relevant section(s) into the relevant Technical Framework volume.

Amend section X.X by the following:

Where the amendment adds text, make the added text **bold underline**. Where the amendment removes text, make the removed text ~~**bold strikethrough**~~. When entire new sections are added, introduce with editor's instructions to "add new text" or similar, which for readability are not bolded or underlined.

General information about IHE can be found at: www.ihe.net.

Information about the IHE Radiology domain can be found at: http://www.ihe.net/Domains/index.cfm.

Information about the organization of IHE Technical Frameworks and Supplements and the process used to create them can be found at: http://www.ihe.net/About/process.cfm and http://www.ihe.net/profiles/index.cfm.

The current version of the IHE Radiology Technical Framework can be found at: http://www.ihe.net/Technical_Framework/index.cfm.

CONTENTS

Introduction to this Supplement

This profile describes the Management of Radiology Report Templates (MRRT). It specifies a data model for templates that will serve as a format for how templates can be transmitted between systems. It also specifies the desirable and common template features that reporting applications may support.

Open Issues and Questions

1. The profile currently specifies that the `src` attribute of an `embed` element shall contain the embedded template's OID concatenated with "`.html`". This facilitates easy rendering of embedded templates by assigning the embedded template a file name matching its OID. Should the embed statement use just the OID string instead?
2. Should the `type` attribute of an `embed` element contain a more specialized `type` than "`text/html`", such as "`IMAGE_REPORT_TEMPLATE`". If so, IHE will need to apply for a new type.

Closed Issues

1. We reviewed RFD and XForms as possible alternative to HTML5 and they were judged less appropriate. HTML5 (http://www.w3.org/TR/html51/) was selected because, like radiology templates, it strikes a balance between expression of coded content and description of a user interface and the related data capture methods. HTML5 is increasingly used as a method for displaying clinical images, facilitating the construction of multi-media reports in the future.
2. The subset of HTML5 used shall be valid XML (e.g., the appropriate close tags shall be used).
3. What are CDA limits on section recursion? None.
4. An interoperable mechanism of retrieval of Merge fields is out of scope.
5. How should applications handle associated text, which when edited, calls into question associated coded entries (e.g., when the user negates text that is semantically linked to a field)? Clarifying text added to Section 8.1.6.3 describing possible application behavior in response to edited text with associated coded content. Span tag mechanism makes association between coded template text and coded content clear.
6. What are the semantics of inheritance among templates? What other relationships should be specified between templates, if any? This functionality will be handled by inclusion of templates in other templates. Only specified relationship is "deprecated by". See Table XXXX

7. Should templates be used to validate report instances? Are there any compelling use cases for this functionality? No compelling use cases were presented. In any case, this would be better addressed using report instance templates, not report authoring templates described in this profile.
8. We have not identified any security issues. Are there any? We have addressed HTTPS in the profile. Because we included SSL, the profile security issues are mitigated. The remaining security issues are out of scope, and relate to report instances that may contain PHI, or merge field access.
9. The draft profile requires a header (title) for each section. Are there use cases where this is a bad idea? Closed. No use cases were presented.
10. Is there a need to have a different section or template attribute to represent the name (rather than "title" or "header")? No use cases were presented.
11. Should the coded content in the template attributes section follow the existing CDA format, essentially serving as a mechanism to pass coded content to a report instance? Yes. There is no other practical mechanism to express coded content using HTML tags.
12. Should the Report Template Creator be able to retrieve templates and if so for what use case? Yes. Retrieve transaction added as optional for Template Creator. Use case is to edit or include existing templates.
13. Should caching for the Retrieve Imaging Report Template transaction be forbidden? No. Profile remains silent on this issue.
14. The Store Imaging Report Template transaction proposes the use of HTTP PUT. Is this the appropriate service? Should a POST be used instead? Are the proposed parameters for the transaction appropriate? What response codes are necessary? Closed. A POST should not be used. Because we can update a template with header information, PUT is appropriate. Parameters and response codes have been revised.
15. What should we call the templates described in this profile to distinguish them from CDA report instance templates? Report authoring templates versus report instance templates. The first paragraph was modified to clarify this distinction.
16. Should correspondence be enforced between `title` in the `head` element and the `dc:title` in the Dublin Core? Yes. Clarified in profile text.
17. Is there a value in having more specific dates than the Dublin Core offers, (e.g., creation date, modification date, release date)? Because each new template version must have a new UID, each unique template does not have a life cycle. Closed. These dates express work flow information related to report template creation. This is best stored and managed elsewhere, rather than in the template itself.
18. Should we manage versioning and lifecycle of templates, and if so, how? No. However, the profile provides a template attribute that signifies the lifecycle stage of the template:

DRAFT, ACTIVE, RETIRED. Applications could extend this formalism based on site preferences.

19. What are the use cases for identifying users and provider groups who are the intended users of a template? Closed. This information is important to store in the templates themselves so it can be queried easily and migrated easily between systems.
20. In the Query Templates transaction, should the Responder simply return entire templates? No. It should only return template metadata so the Requester can choose the template(s) necessary, to conserve bandwidth / processing.
21. Is the ability to trigger a template by selecting a menu item (Section 8.1.4) necessary, sufficient, or excessive? Necessary but not sufficient. An inclusion mechanism is described so templates can be included by reference into other templates, even when not in response to selection of a menu item.
22. How should querying for coded values work? Is specifying the coding system required? If so, how should it be specified? For example, within the same parameter value separated by a colon? I.e., codingValue= RADLEX:RID6434 as opposed to codingSystem=RADLEX&codingValue=RID6434. This has been clarified in the text.
23. Should the Query Templates transaction explain more succinctly how AND and OR operations work? Should the lowerDate and upperDate logic be more thoroughly described? This has been clarified in the text.
24. Should it be possible to include templates in other templates by reference? Yes. This creates a potential conflict of namespaces between the two templates. To resolve the conflict, identifiers in the embedded template shall be prefixed with the **`title`** element of the embedded template.

General Introduction

Update the following Appendices to the General Introduction as indicated below. Note that these are not appendices to Volume 1.

Appendix A - Actor Summary Definitions

Modify Report Creator and add the following two actors to the IHE Technical Frameworks General Introduction list of Actors:

Actor	Definition
Report Creator	A system that generates and transmits draft (and optionally, final) diagnostic reports~~, **presenting them as DICOM Structured Reporting Objects**~~. It may also retrieve work list entries for reporting steps from the Report Manager and provide notification of completion of the step, allowing the enterprise to track the status of an awaited report.
Report Template Manager	A system that provides storage and management of report templates.
Report Template Creator	A system that enables a user to create and edit report templates.

Appendix B - Transaction Summary Definitions

Add the following transactions to the IHE Technical Frameworks General Introduction list of Transactions:

Transaction	Definition
Retrieve Imaging Report Template [RAD-103]	A Requester retrieves a template from a Responder.
Store Imaging Report Template [RAD-104]	A Sender stores a report template to a Receiver.
Query Imaging Report Templates [RAD-105]	A Requester queries for a list of templates from a Responder.

Glossary

Add the following glossary terms to the IHE Technical Frameworks General Introduction Glossary:

Glossary Term	Definition
none	

Volume 1 – Profiles

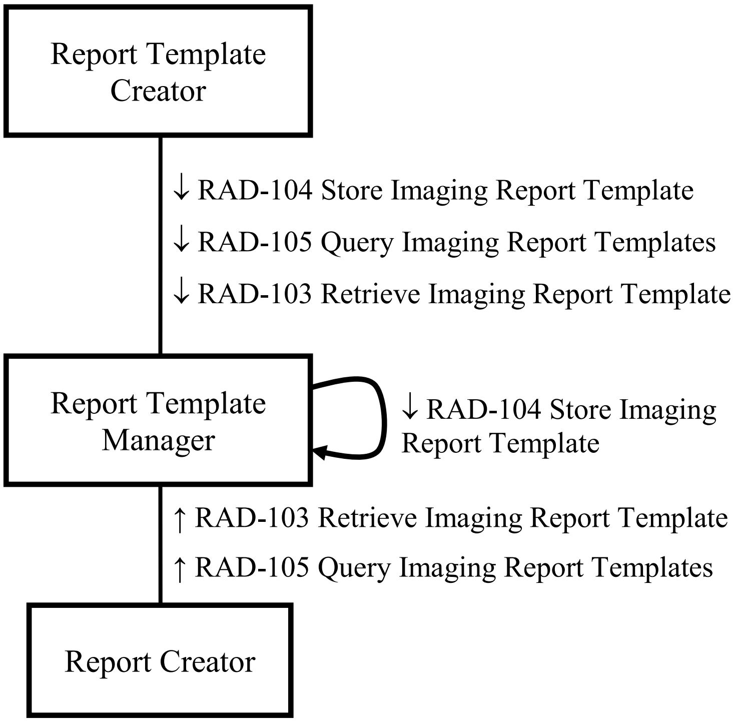
33 Management of Radiology Report Templates (MRRT) Profile

This workflow profile concerns the use of imaging report templates to create diagnostic imaging reports. This profile distinguishes between *report authoring templates*, which are templates used by radiologists to guide the creation of a clinical imaging report, and *report instance templates*, which describe technical constraints on the structure and content of imaging report instances, such as the constraints described in and XML schema or the HL7 Clinical Document Architecture. This profile applies to the former, *report authoring templates,* and describes methods for the formatting of imaging report templates and the management of their transport between template libraries and report creation systems. Specifically, this profile describes an enhanced feature set for report authoring templates, delineates how such templates from a vendor-agnostic template library could be used immediately by a reporting system, and provides a format for migration of templates between reporting systems.

33.1 MRRT Actors, Transactions, and Content Modules

This section defines the actors, transactions, and content modules in this profile. General definitions of actors are given in the Technical Frameworks General Introduction Appendix A at http://www.ihe.net/Technical_Framework/index.cfm.

Figure 33.1-1 shows the actors directly involved in the MRRT Profile and the relevant transactions between them. If needed for context, other actors that may be indirectly involved due to their participation in other related profiles are shown in dotted lines. Actors that have a mandatory grouping are shown in conjoined boxes.


Report Template Creator
↓ RAD-104 Store Imaging Report Template
↓ RAD-105 Query Imaging Report Templates
↓ RAD-103 Retrieve Imaging Report Template
Report Template Manager
↓ RAD-104 Store Imaging Report Template
↑ RAD-103 Retrieve Imaging Report Template
↑ RAD-105 Query Imaging Report Templates
Report Creator

285

Figure 33.1-1: MRRT Actor Diagram

Table 33.1-1 lists the transactions for each actor directly involved in the MRRT Profile. To claim compliance with this Profile, an actor shall support all required transactions (labeled "R") and may support the optional transactions (labeled "O").

290

Table 33.1-1: MRRT Profile - Actors and Transactions

Actors	Transactions	Optionality	Reference
Report Template Manager	Retrieve Imaging Report Template [RAD-103]	R	RAD TF-3: 4.103
	Store Imaging Report Template [RAD-104]	R	RAD TF-3: 4.104
	Query Image Report Templates [RAD-105]	R	RAD TF-3: 4.105
Report Creator	Retrieve Imaging Report Template [RAD-103]	R	RAD TF-3: 4.103
	Query Image Report Templates [RAD-105]	O	RAD TF-3: 4.105
Report Template Creator	Store Imaging Report Template [RAD-104]	R	RAD TF-3: 4.104

Actors	Transactions	Optionality	Reference
	Retrieve Imaging Report Template [RAD-103]	O	RAD TF-3: 4.103
	Query Image Report Templates [RAD-105]	O	RAD TF-3: 4.105

33.1.1 Actor Descriptions and Actor Profile Requirements

33.1.1.1 Report Template Manager

The Report Template Manager shall support Store Imaging Report Template [RAD-104] as both the Sender and Receiver.

33.2 MRRT Actor Options

Options that may be selected for each actor in this profile, if any, are listed in the Table 33.2-1. Dependencies between options when applicable are specified in notes.

Table 33.2-1: Management of Radiology Report Templates - Actors and Options

Actor	Option Name	Reference
Report Template Manager	No options defined	-
Report Creator	No options defined	-
Report Template Creator	No options defined	-

33.3 MRRT Required Actor Groupings

None

33.4 MRRT Overview

33.4.1 Concepts

A template is a document with a preset structure, used as a starting point for a Report Creator, so that the structure does not have to be recreated each time it is used. A template describes how a Report Creator should interact with a user to create a report instance.

For example, a radiologist may use a speech recognition system to create a narrative report describing the interpretation of a diagnostic imaging study. The speech recognition system assists the radiologist in applying templates during the reporting process. The radiologist also may edit the text of the report using that system.

A previously published IHE whitepaper on the management of radiology report templates provides more detail on how radiologists typically employ imaging report templates.

33.4.2 Use Cases

33.4.2.1 Use Case #1: New Template Storage

33.4.2.1.1 New Template Storage Use Case Description

A radiologist may need a template to help create a report for a specific study. If such a template is not available in an accessible template library, the radiologist may use a Report Template Creator to create, then edit, a template that serves the need. That template is stored in a Report Template Manager where it is available for later retrieval by a Report Creator. In some cases a single application may serve as both Report Template Creator and Report Template Manager. In other cases the Report Template Creator may be a separate application.

33.4.2.1.2 New Template Storage Process Flow

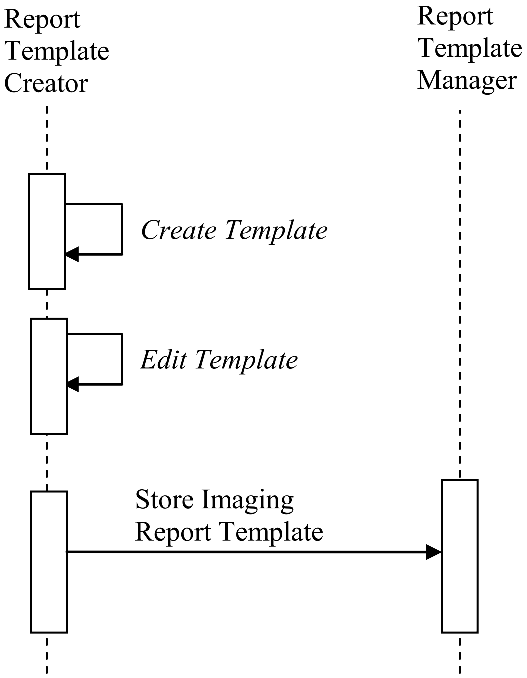


Figure 33.4.2.1.2-1: New Template Storage Process Flow in MRRT Profile

33.4.2.2 Use Case #2: Template Import

33.4.2.2.1 Template Import Use Case Description

A radiologist may need a template to help create a report for a specific study. The Report Creator queries for and retrieves an appropriate template or templates from the Report Template Manager, either based on the users request or an automated algorithm. In some cases a single application may serve as both Report Template Manager and Report Creator, but in other cases the Report Template Manager may be maintained by a radiology practice, a reporting vendor, or a professional group. After selecting the template, the user creates a report based on the template.

33.4.2.2.2 Template Import Process Flow

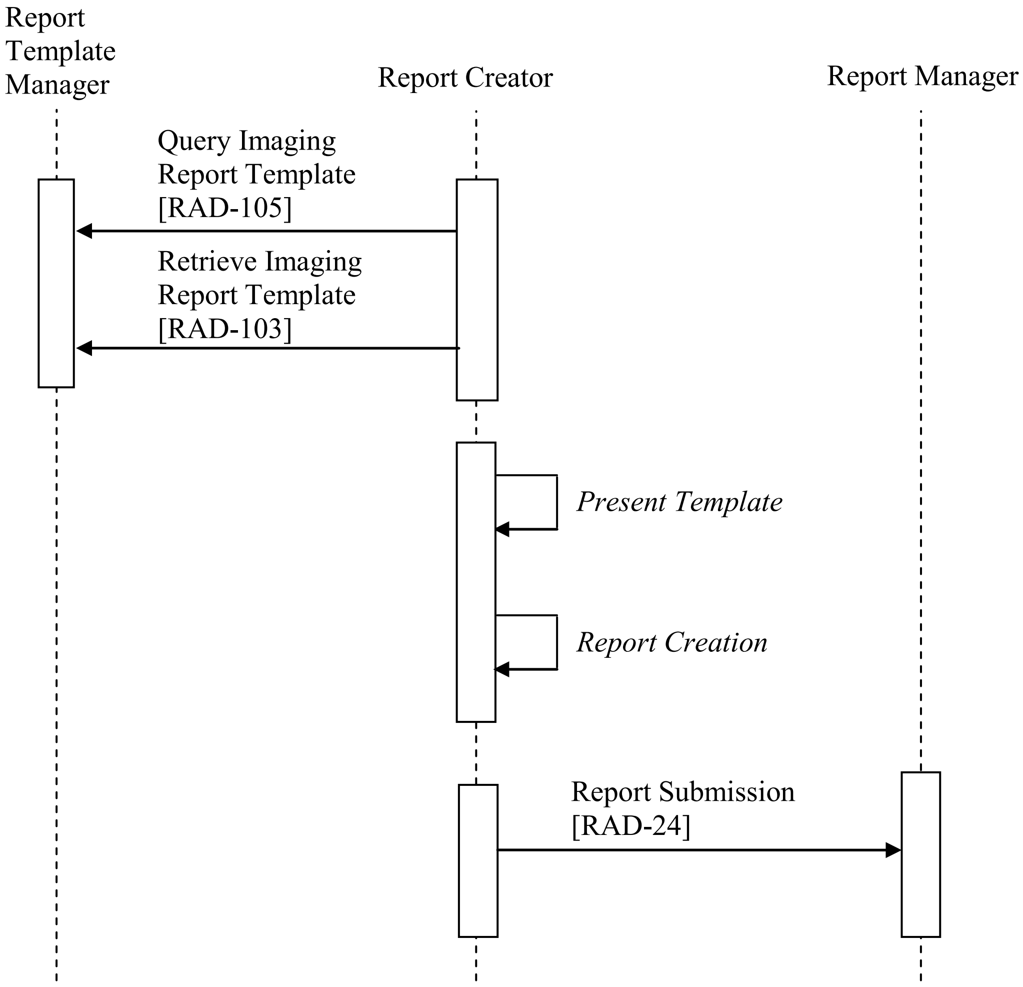

Report Template Manager
Report Creator
Report Manager
Query Imaging Report Template [RAD-105]
Retrieve Imaging Report Template [RAD-103]
Present Template
Report Creation
Report Submission [RAD-24]

Figure 33.4.2.2.2-1: Template Import Process Flow in MRRT Profile

Note: Report Submission [RAD-24] is not part of this profile. It is shown to provide context.

33.4.2.3 Use Case #3: Template Migration

33.4.2.3.1 Template Migration Use Case Description

If a radiology practice elects to change reporting vendors, the practice will need to transmit its templates from their current Report Template Manager to a new Report Template Manager. They would use the Store Imaging Report Template transaction to migrate the templates from their old Report Template Manager to the new Report Template Manager. This eliminates the need for custom programming to reverse-engineer the template format of the old vendor.

33.4.2.3.2 Template Migration Process Flow

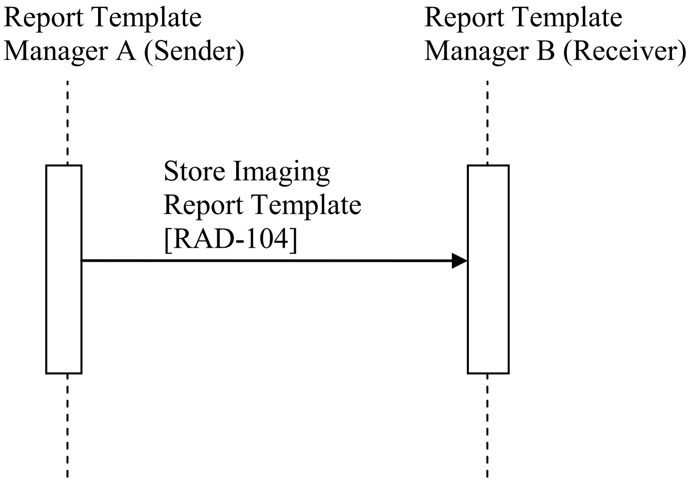


Figure 33.4.2.3.2-1: Template Migration Process Flow in MRRT Profile

33.4.2.4 Use Case #4: Build Template

33.4.2.4.1 Build Template Description

Similar to Use Case #1, a radiologist may need a template to help create a report for a specific study. The radiologist may use a Report Template Creator to create, then edit, a template that serves the need. However, this template utilizes common pieces from other templates. That template is stored in a Report Template Manager where it is available for later retrieval by a Report Creator.

33.4.2.4.2 Template Import Process Flow

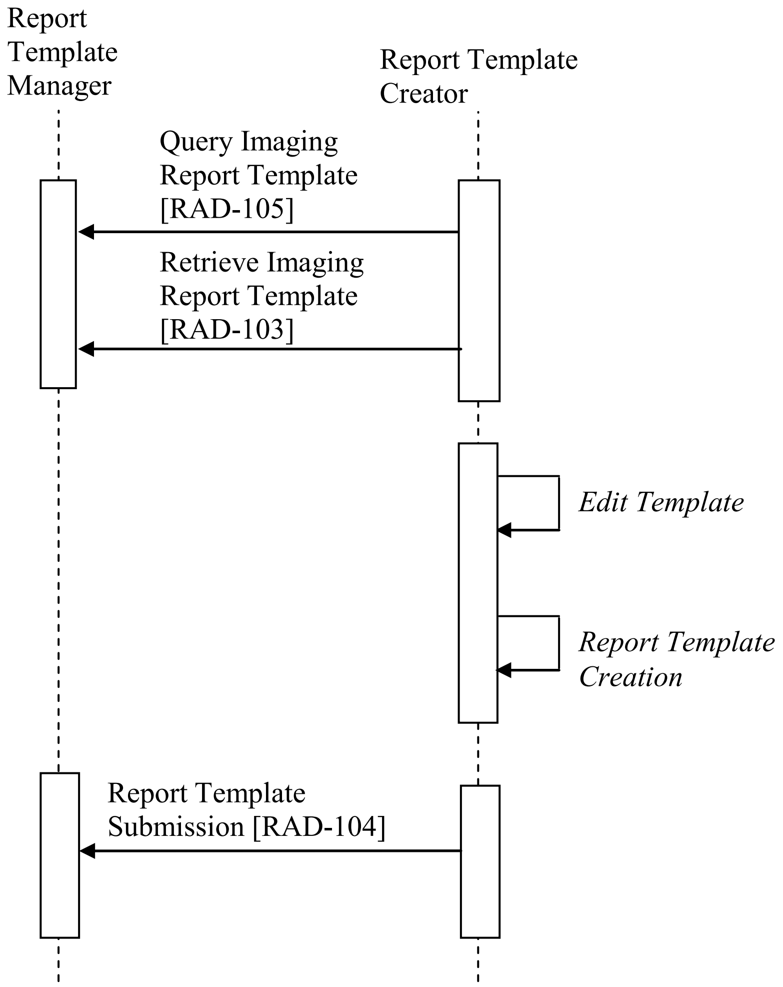

Report Template Manager
Report Template Creator
Query Imaging Report Template [RAD-105]
Retrieve Imaging Report Template [RAD-103]
Edit Template
Report Template Creation
Report Template Submission [RAD-104]

Figure 33.4.2.4.2-1: Build Template Process Flow in MRRT Profile

360 33.4.2.5 Use Case #5: Manage Template

33.4.2.5.1 Manage Template Description

A radiologist has decided that a particular template should be retired, due to replacement to a new template. The radiologist may use a Report Template Creator to edit a template that changes the metadata. That template is stored in a Report Template Manager where it available for later retrieval by a Report Creator.

33.4.2.5.2 Manage Template Process Flow

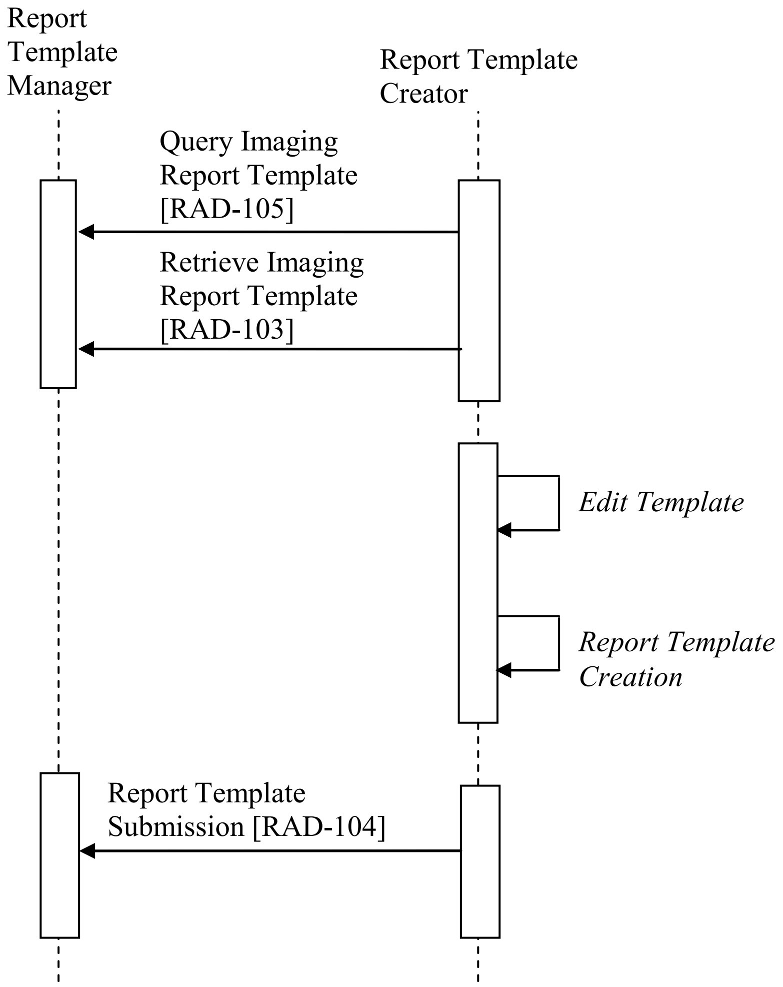

Report Template Manager
Report Template Creator
Query Imaging Report Template [RAD-105]
Retrieve Imaging Report Template [RAD-103]
Edit Template
Report Template Creation
Report Template Submission [RAD-104]

Figure 33.4.2.5.2-1: Manage Template Process Flow in MRRT Profile

370 33.5 MRRT Security Considerations

None

33.6 MRRT Cross Profile Considerations

SINR - Simple Image and Numeric Report. A Report Creator in SINR might be grouped with a Report Template Manager in this profile, yielding a unified reporting application that manages its own templates.

The transactions outlined here generally will occur before the transactions described in SINR and shown in RAD TF-1: Figure 9.3-1.

Volume 3 – Transactions (continued)

4.103 Retrieve Imaging Report Template [RAD-103]

4.103.1 Scope

This transaction is used to retrieve a template from a Report Template Manager in the proper format.

4.103.2 Actor Roles

Table 4.103.2-1: Actor Roles

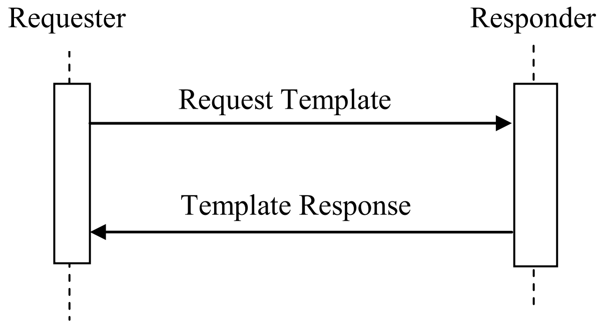
Role:	**Requester:** Requests a template or templates from the Responder
Actor(s):	The following Actors may play the role of Requester: • Report Creator
Role:	**Responder:** Provides a template or templates in response to the request
Actor(s):	The following Actors may play the role of Responder: • Report Template Manager

Transaction text specifies behavior for each Role. The behavior of specific Actors may also be specified when it goes beyond that of the general Role.

4.103.3 Referenced Standards

- IETF RFC2616 HyperText Transfer Protocol HTTP/1.1
- Extensible Markup Language (XML) 1.0 (Second Edition). W3C Recommendation 6 October 2000. http://www.w3.org/TR/REC-xml.
- Dublin Core Metadata Element Set, standardized as ISO Standard 15836: 2009 and ANSI/NISO Standard Z39.85-2012. http://dublincore.org/documents/dces/

4.103.4 Interaction Diagram



395

Figure 4.103.4-1: Interaction Diagram for Retrieve Imaging Report Template [RAD-103]

4.103.4.1 Request Template Message

The Requester sends a message to the Responder indicating the templates it would like to receive.

400 A Responder shall support handling such messages from more than one Requester. A Requester may choose to support making requests to more than one Responder.

4.103.4.1.1 Trigger Events

1. A Requester needs to collect templates for later use in anticipation of reporting in the future.
2. 405 The user of a Requester, such as a Report Creator, invokes a template.
3. A Requester needs to retrieve a template that has been referenced by another template.

4.103.4.1.2 Message Semantics

The message is an HTTP GET request.

The HTTP request shall include the following parameters to identify the template to be returned. 410 All parameter names and values are case-sensitive.

Table 4.103.4.1.2-1: HTTP Path Parameters

Parameter Name	REQ	Description	Values
`templateUID`	R	Identifies template's UID as known to both actors, expressed by the dc:identifier element shown in Table 8.1.1-1.	This value shall be a properly defined Object identifier (OID) as specified in ITI TF-2x: Appendix B.

The only binding required for both the Requester and Responder is the binding to the HTTP-GET. In this binding the sample message will be formatted as follows:

```
http://<location>/IHETemplateService/<templateUID>
```

The **`<location>`** part of the URL shall contain the host name, an optional port address, and may be followed by an optional path. The remainder of the URL, including **`IHETemplateService`** and the following request parameters may not be changed. See the discussion about location in ITI TF-2a: 3.11.4.1.2 Message Semantics.

If necessary, the Requester may perform the request to the web service utilizing HTTPS protocol. The Responder shall respond using HTTPS if requested.

The Responder may return HTTP redirect responses to a request. The Requester can expect to receive an error response, or the data requested, or a request to look elsewhere for the data. The Requester shall follow redirects, but if a loop is detected, it may report an error.

4.103.4.1.3 Expected Actions

The Responder shall parse the request and create a response containing the templates meeting the parameters of the request in the proper format. If multiple requests are received, each is handled in sequence.

The Responder shall provide a response message header containing the appropriate status code indicating success, warning, or failure as shown in table 4.103.4.1.3-1.

Table 4.103.4.1.3-1: HTTP Responses

Service Status	HTTP1.1 Status Codes	Description
Failure (see note 2)	503 – Busy	This indicates that the Responder was unable to provide the template because it was out of resources.
	404 – Not Found	This indicates that the Responder was unable to provide the template because it did not exist on the responder at the time of the request.
	401 - Unauthorized	This indicates that the Responder refused to provide a template because authentication credentials were not provided or not sufficient.
	400 – Bad Request	This indicates that the Responder was unable to provide the template because the template UID is missing or corrupt.
Success	200 – OK	This indicates that the request was successful and the Responder will provide the template.

Note 1: Other HTTP response codes may be returned by the Responder, indicating conditions outside of the scope of this transaction.

Note 2: It is recommended that the Responder complement the returned error code with a human readable description of the error condition.

If an error condition cannot be automatically recovered, at a minimum, the error should be displayed to the user by the Requester.

The Requester may wish to request any templates that are embedded in the response (see section 8.1.4) immediately, rather than retrieve embedded templates on demand later when the Responder may not be available.

4.103.4.2 Template Response Message

The Responder transmits the requested templates to the Requester.

4.103.4.2.1 Trigger Events

The Template Response message is created in response to a Responder receiving a Request Template message.

4.103.4.2.2 Message Semantics

The message is a document in a HTTP GET response.

4.103.4.2.3 Expected Actions

The Responder shall format the document according to content definition in RAD TF-3: 8.1, and return it in the HTTP response. The document shall be processed according to the features, configuration, and business logic of the Requester. Possibilities include making the template accessible to the user.

4.103.5 Security Considerations

Although the content of templates is not typically protected information, for consistency with other transactions on the client, which likely will involve protected information, it is reasonable to expect support for HTTPS.

4.103.5.1 Security Audit Considerations

None

4.104 Store Imaging Report Template [RAD-104]

4.104.1 Scope

This transaction is used to store templates in the proper format on another system.

4.104.2 Actor Roles

Table 4.104.2-1: Actor Roles

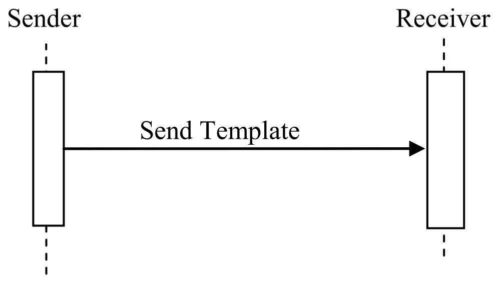
Role:	**Sender:** Sends and requests storage of templates
Actor(s):	The following Actors may play the role of Sender: • Report Template Creator • Report Template Manager
Role:	**Receiver:** Receives and stores templates
Actor(s):	The following Actors may play the role of Receiver: • Report Template Manager

465 Transaction text specifies behavior for each Role. The behavior of specific Actors may also be specified when it goes beyond that of the general Role.

4.104.3 Referenced Standards

- IETF RFC2616 HyperText Transfer Protocol HTTP/1.1
- Extensible Markup Language (XML) 1.0 (Second Edition). W3C Recommendation 6 October 2000. http://www.w3.org/TR/REC-xml. 470
- Dublin Core Metadata Element Set, standardized as ISO Standard 15836: 2009 and ANSI/NISO Standard Z39.85-2012. http://dublincore.org/documents/dces/

4.104.4 Interaction Diagram



475 **Figure 4.104.4-1: Interaction Diagram for Store Imaging Report Template [RAD-104]**

4.104.4.1 Send Template Message

The Sender provides the template to a Receiver.

A Receiver shall support handling such messages from more than one Sender. A Sender may choose to support storing templates to more than one Receiver.

4.104.4.1.1Trigger Events

1. A Report Template Manager (acting as a Sender) needs to transmit a template to another Report Template Manager (acting as a Receiver) for storage.
2. A Report Template Creator (acting as a Sender) needs to store a template that it has created.

4.104.4.1.2 Message Semantics

The message is an HTTP PUT request. The Sender shall format the document according to content definition in RAD TF-3: 8.1.

The HTTP request shall include the following parameters to identify the template to be stored. All parameter names and values are case-sensitive.

Table 4.104.4.1.2-1: HTTP Path Parameters

Parameter Name	REQ	Description	Values
`templateUID`	R	Identifies template's UID as known to both actors.	This value shall be a properly defined Object identifier (OID) as specified in ITI TF-2x: Appendix B.

The only binding required for both the Sender and Receiver is the binding to the HTTP-PUT. In this binding the sample message will be formatted as follows:

```
http://<location>/IHETemplateService/<templateUID>
```

The **`<location>`** part of the URL shall contain the host name, an optional port address, and may be followed by an optional path. The remainder of the URL, including **`IHETemplateService`** and the following request parameters may not be changed.

If necessary, the Sender may perform the request to the web service utilizing HTTPS protocol. In this case, the Receiver shall respond using HTTPS.

The Receiver may return HTTP redirect responses to a request. The Sender can expect to receive an error response, or the data requested, or a request to look elsewhere for the data. The Sender shall follow redirects, but if a loop is detected, it may report an error.

4.104.4.1.3 Expected Actions

The Receiver shall accept the request to store the template. If multiple requests are received, each is handled in sequence.

If the unique ID of the template already exists on the Receiver, the Receiver shall replace the existing template with the received template. Replacement of templates is intended to allow meta-data to be updated (for example setting the Status to RETIRED) but is not intended to permit modification of the template content itself. If the template content was modified, the Template Creator will have assigned a new OID.

The Receiver shall provide a response message header containing the appropriate status code indicating success, warning, or failure as shown in Table 4.104.4.1.3-1.

Table 4.104.4.1.3-1: HTTP Responses

Service Status	HTTP1.1 Status Codes	Description
Failure (see note 2)	503 – Busy	This indicates that the Responder was unable to store the template because it was out of resources.
	422 – Unprocessable Entity	This indicates that the Responder was unable to store the template because the template does not conform to RAD TF-3: 8.1.
	401 - Unauthorized	This indicates that the Responder refused to store the template because authentication credentials were not provided or not sufficient.
	400 – Bad Request	This indicates that the Responder was unable to store the template because the template UID is missing or corrupt.
Success	200 – OK	This indicates that the request was successful and the Responder has stored the template.

Note 1: Other HTTP response codes may be returned by the Receiver, indicating conditions outside of the scope of this transaction.

Note 2: It is recommended that the Receiver complement returned error code with a human readable description of the error condition.

If an error condition cannot be automatically recovered, at a minimum, the error should be displayed to the user by the Sender.

4.104.5 Security Considerations

Although the content of templates is not typically protected information, for consistency with other transactions on the client, which likely will involve protected information, it is reasonable to expect support for HTTPS.

4.104.5.1 Security Audit Considerations

None

4.105 Query Imaging Report Template [RAD-105]

4.105.1 Scope

This transaction is used to query templates from a Report Template Manager in the proper format.

4.105.2 Actor Roles

Table 4.105.2-1: Actor Roles

Role:	**Requester:** Requests a filtered list of templates from the Responder
Actor(s):	The following Actors may play the role of Requester: • Report Creator
Role:	**Responder:** Provides a list of templates in response to the request
Actor(s):	The following Actors may play the role of Responder: • Report Template Manager

Transaction text specifies behavior for each Role. The behavior of specific Actors may also be specified when it goes beyond that of the general Role.

535 ## 4.105.3 Referenced Standards

- IETF RFC2616 HyperText Transfer Protocol HTTP/1.1
- Extensible Markup Language (XML) 1.0 (Second Edition). W3C Recommendation 6 October 2000. http://www.w3.org/TR/REC-xml.
- Dublin Core Metadata Element Set, standardized as ISO Standard 15836: 2009 and ANSI/NISO Standard Z39.85-2012. http://dublincore.org/documents/dces/ 540

4.105.4 Interaction Diagram

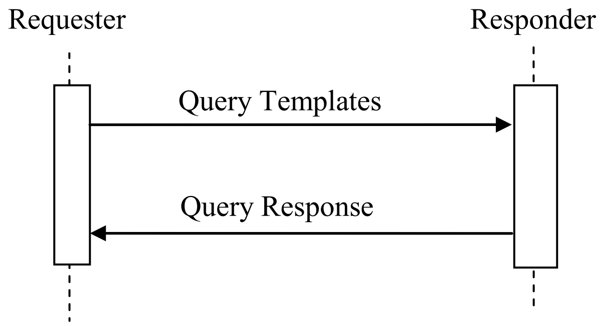


Figure 4.105.4-1: Interaction Diagram for Query Imaging Report Templates [RAD-105]

4.105.4.1 Query Templates Message

545 The Requester sends a message to the Responder indicating the list of templates it would like to receive.

A Responder shall support handling such messages from more than one Requester. A Requester may choose to support making requests to more than one Responder.

4.105.4.1.1 Trigger Events

1. A Requester needs to collect templates for later use in anticipation of selecting an appropriate template for reporting in the future.
2. The user of a Requester, such as a Report Creator, invokes a template query.

4.105.4.1.2 Message Semantics

The message is an HTTP GET request.

To filter the template matches to be returned, the HTTP request shall include one or more of the following parameters. All parameter names and values are case-sensitive.

All parameters shall be supported by the Responder, and are optional for the Requester.

Table 4.105.4.1.2-1: HTTP Query Parameters

Parameter Name	REQ	Description	Values
`title`	O	Wildcard query of the **`dc:title`** tag.	This value shall be a string.
`identifier`	O	Exact query of the **`dc:identifier`** tag.	This value shall be a properly defined Object identifier (OID) as specified in ITI TF-2x: Appendix B.
`creator`	O	Wildcard query of the **`dc:creator`** tag.	This value shall be a string.
`publisher`	O	Wildcard query of the **`dc:publisher`** tag.	This value shall be a string.
`license`	O	Wildcard query of the **`dc:license`** tag.	This value shall be a string.
`lower_date`	O	Query the date of the template for values on or after the specified lower date. See Note 1.	This value shall be encoded in the XML primitive date format.
`upper_date`	O	Query the date of the template for values on or before the specified upper date. See Note 1.	This value shall be encoded in the XML primitive date format.
`language`	O	Wildcard query of the **`dc:language`** tag.	This value shall be an ISO 639 two-letter language code.
`top_level_flag`	O	Exact query of the **`top-level-flag`** tag.	This value shall be an xsd:Boolean.
`status`	O	Exact query of the **`status`** tag.	This value shall be a string.
`code_value`	O	Exact query of the entry/term/ **`coding_scheme_designator`** and **`code_value`** tags.	This value shall be a string containing the coding scheme from which the code value was drawn, and the code value itself separated by a colon. See

Parameter Name	REQ	Description	Values
			Note 2.
`code_meaning`	O	Wildcard query of the entry/term/code_meaning tag.	This value shall be a string.
`limit`	O	Limits the results returned to a maximum number. If omitted, all matching results shall be returned.	This value shall be an integer.
`offset`	O	Skips the first number of matching results. If omitted, no results will be skipped.	This value shall be an integer.
`sort`	O	Returns the results in alphabetical order of a specified field. If omitted, results are ordered by title.	This value shall be a string, being one of the query parameters aside from limit, offset, or sort.

Note 1: Lower_date and upper_date are used to constrain the date of the template. For example, a lower_date of 2010-01-01 and an upper_date of 2010-12-31 will return all templates with a date in the year 2010. The query is inclusive of the date specified. If the value passed is not in XML primitive date format, an HTTP 400 error will be returned.

Note 2: When searching for code value, the coding system must also be specified. This is done by concatenating the coding system, a colon, and the code value. For example, when searching in RadLex (2.16.840.1.113883.6.256) for Computed Tomography (RID10321), the code_value to be queried for would be "2.16.840.1.113883.6.256:RID10321".

For the parameters specified as wildcard search, a template matches if the text string in the parameter value appears in the corresponding attribute of the template. Wildcard matching is insensitive to case. For example, searching the title for "abdomen" would match templates with "CT Abdomen".

Specifying multiple, different parameters indicates an AND relation, meaning that both parameter values must be present for the template to match. For example, requesting that the title contains "CT" and the publisher contains "Hospital" would return only templates that met both of those criteria. However, specifying the same parameter multiple times indicates an OR relation, where only one of the parameter values must be present in the template. For example, requesting that the title contains "CT" or the title titles "US" would return templates that match either of those criteria. For more complicated searches, it is expected that the client will be capable of retrieving a large set of templates and performing further filtering internally.

The only binding required, for both the Requester and Responder, is the binding to the HTTP-GET. In this binding the sample message will be formatted as follows:

```
http://<location>/IHETemplateService/?[parameter1_name]=[parameter1 value]&
[parameter2_name]=[parameter2 value]
```

The **`<location>`** part of the URL shall contain the host name, an optional port address, and may be followed by an optional path. The remainder of the URL, including **`IHETemplateService`** and the following request parameters may not be changed. See the discussion about location in ITI TF-2a: 3.11.4.1.2 Message Semantics.

If no search parameters are provided, all template headers known by the Responder shall be returned.

If necessary, the Requester may perform the request to the web service utilizing HTTPS protocol. The Responder shall respond using HTTPS if requested.

The Responder may return HTTP redirect responses to a request. The Requester can expect to receive an error response, or the data requested, or a request to look elsewhere for the data. The Requester shall follow redirects, but if a loop is detected, it may report an error.

4.105.4.1.3 Expected Actions

The Responder shall parse the request and create a response containing the template headers meeting the parameters of the request in the proper format. If multiple requests are received, each is handled in sequence.

The Responder shall provide a response message header containing the appropriate status code indicating success, warning, or failure as shown in Table 4.105.4.1.3-1.

Table 4.105.4.1.3-1: HTTP Responses

Service Status	HTTP1.1 Status Codes	Description
Failure (see note 2)	503 – Busy	This indicates that the Responder was unable to perform the query because it was out of resources.
	401 - Unauthorized	This indicates that the Responder refused to return results because authentication credentials were not provided or not sufficient.
	400 – Bad Request	This indicates that the Responder was unable to provide the template list because one or more of the parameters are corrupt.
Success	200 – OK	This indicates that the request was successful and the Responder will list matching templates.

Note 1: Other HTTP response codes may be returned by the Responder, indicating conditions outside of the scope of this transaction.

Note 2: It is recommended that the Responder complement the returned error code with a human readable description of the error condition.

If an error condition cannot be automatically recovered, at a minimum, the error should be displayed to the user by the Requester.

4.105.4.2 Template Response Message

The Responder transmits the requested template headers to the Requester.

4.105.4.2.1 Trigger Events

The Templates Response message is created in response to a Responder receiving a Query Templates message.

4.105.4.2.2 Message Semantics

The message is a document according to content definition in RAD TF-3: 8.2 in a HTTP GET response.

4.105.4.2.3 Expected Actions

The Responder shall return all matches in the HTTP response.

4.105.5 Security Considerations

Although the content of templates is not typically protected information, for consistency with other transactions on the client, which likely will involve protected information, it is reasonable to expect support for HTTPS.

4.105.5.1 Security Audit Considerations

None

Volume 3 – Content Modules

8.1 Report Template Structure

The format for report templates is expressed in HTML5 with extensions. Some of the extensions, such as custom data attributes, are supported by HTML5, while others, such as coded content in XML format, are not. Whenever possible, existing HTML5 tags are used to express template content. All HTML5 tags shall be closed so the report template can be validated as XML. Except for two XML blocks at the end of the report template, the document can be validated using widely available HTML5 tools. Internal and external CSS style sheets may be used to render these templates. Inline styles are not permitted.

Although the format for report templates is expressed in HTML5, the Report Creator is not required to use an HTML5 rendering engine. HTML is simply a convenient method to express the concepts in the template. The Report Creator is required to expose the appropriate behavior, for example allow the user to select an item from a selection list.

Examples of template content are provided in the Figures below. In addition, an example template and external style sheet are available on the IHE ftp site.

Each template shall support the constraints described below. The template:

1. Shall begin with exactly one [1..1] HTML5 **`DOCTYPE`** declaration: **`<!DOCTYPE html>`**
2. Shall contain exactly one [1..1] **`html`** element.
    a. The **`html`** element shall contain exactly one [1..1] **`head`** element.
        i. Identifiers in the **`head`** element shall use an underscore (“_”) as separator.
        ii. The **`head`** element shall contain exactly one [1..1] **`title`** element containing the name of the template. The value of the **`title`** element shall be the same as the **`dc:title`** element shown in Table 8.1.1-1.
        iii. The **`head`** element shall contain exactly one [1..1] **`meta`** element declaring the character set used: **`<meta charset="UTF-8">`**.
        iv. The **`head`** element may contain zero or more [0..*] **`meta`** elements encoding Dublin core attributes of the template, as shown in Figure 8.1-1.
            1. The **`name`** property of the **`meta`** element will be used to specify the template attribute.
            2. For Dublin Core template attributes, the “**`dcterms`**” namespace shall be used.
            3. The **`content`** property of the **`meta`** element will be used to specify the value of the template attribute.

```
   <head>
      <title>CT Brain</title>
      <meta charset="UTF-8">
      <meta name="dcterms.title" content="CT Brain">
      <meta name="dcterms.identifier" content="1.2.3.4.5">
      <meta name="dcterms.type" content="IMAGE_REPORT_TEMPLATE">
      <meta name="dcterms.language" content="en">
      <meta name="dcterms.publisher" content="Radiological Society of North
America (RSNA)">
      <meta name="dcterms.rights" content="May be used freely, subject to
license agreement">
      <meta name="dcterms.license"
content="http://www.radreport.org/license.pdf">
      <meta name="dcterms.date" content="2012-03-28">
      <meta name="dcterms.creator" content="Flanders AE, et al.">
      <meta name="dcterms.contributor" content="Bozkurt S [coder]">
      <meta name="dcterms.contributor" content="Kahn CE Jr [editor]">
      <meta name="dcterms.contributor" content="American Society of
Neuroradiology (ASNR)">
      <link rel="stylesheet" type="text/css" href="IHE_Template_Style.css" />
      <!--non-Dublin-Core coded content here-->
   </head>
```

Figure 8.1-1: Example of Dublin Core coded content in the `<head>` element

v. The **`head`** element shall contain exactly one [1..1] **`script`** element containing coded content.

   1. The **`script`** element shall be assigned a **`type`** attribute of “**`text/xml`**”.

   2. The **`script`** element shall contain exactly one [1..1] **`template_attributes`** element.

      a. The **`template_attributes`** element may contain zero or more [0..*] elements containing additional coded content applicable to the entire template that cannot be represented in Dublin Core attributes.

      b. **`term`** elements may be assigned a **`type`** attribute indicating the template attribute for which the term tuple is the value. See the example shown in Figure 8.1-2.

```
   <head>
      <!--title and Dublin Core meta elements here-->
      <script type="text/xml">
         <template_attributes>
            <top_level_flag>true</top_level_flag>
            <term type="modality">
```

```
                  <code_meaning>computed tomography</code_meaning>
                  <code_value>RID10321</code_value>

   <coding_scheme_designator>2.16.840.1.113883.6.256</coding_scheme_designato
r>
            </term>
            <term type="body part">
                  <code_meaning>brain</code_meaning>
                  <code_value>RID6434</code_value>

   <coding_scheme_designator>2.16.840.1.113883.6.256</coding_scheme_designato
r>
            </term>
         </template_attributes>
      <!--Other coded content here-->
      </script>
   </head>
```

Figure 8.1-2: Example of non-Dublin-Core template attributes in the `<script>` element

3. The `script` element shall contain exactly one [1..1] `coded_content` element, such as that shown in Figure 8.1-3.
    a. Coded content shall be expressed as described in 8.1.6.1.
    b. Identifiers in the `coded_content` section shall use underscore ("_") as a separator to maintain compatibility with XML and HL7 formats.
    c. The `coded_content` element may contain zero or more [0..*] `entry` elements.
    d. The opening tag of each `entry` element shall contain an `ORIGTXT` attribute whose value matches the `id` attribute of an element in the `body` to which the coded content in the `entry` applies.

```
   <head>
      <!-- title and Dublin Core meta elements here -->
      <script type="text/xml">
         <!--template_attributes element here-->
         <coded_content>
            <entry ORIGTXT="T003">
               <term>
                  <code_meaning>Ankle</code_meaning>
                  <code_value>RID28545</code_value>
                  <coding_scheme_designator>RadLex</coding_scheme_designator>
               </term>
               <term>
                  <code_meaning>Right</code_meaning>
                  <code_value>RIC5825</code_value>
                  <coding_scheme_designator>RadLex</coding_scheme_designator>
               </term>
            </entry>
         <!—additional entry elements here-->
         </coded_content>
      </script>
   </head>
```

Figure 8.1-3: Coded Content Example

vi. The **head** element may contain style information formatted according to HTML5 standards, using the **style** element for internal CSS style elements and the **link** element for CSS files.

vii. The **head** element shall comply with all other HTML5 constraints.

b. The **html** element shall contain exactly one [1..1] **body** element.

i. Identifiers in the **body** element shall use a hyphen ("-") as separator to maintain compatibility with HTML5 features, such as custom coded content.

ii. The **body** element shall contain at least one [1..*] **section** element whose opening tag specifies the attributes described in Table 8.1.2-1.

1. Each **section** element shall contain exactly one [1..1] **header** element.

a. The opening tag of the **header** element shall contain exactly one [1..1] **class** attribute indicating the section level. The value of the attribute shall be the string "level" followed by an integer indicating the nesting level (e.g., "level1").

b. The header element may contain the title text for the section.

2. Each **section** element shall contain at least one [1..*] HTML paragraph (**p**) element containing the section content.

775 ## 8.1.1 Template Attributes

The following Dublin Core metadata attributes may be associated with each report template in the **`template_attributes`** element.

Table 8.1.1-1: Dublin Core Metadata Elements for Report Templates

Dublin Core Template Metadata Elements	Description	Vocabulary Constraint	Opt
`dc:title`	A human readable name for the template. There is enforced correspondence with the title element in the head.	This value shall be the same as the value of the **`title`** element of the **`head`** element RAD TF-3: 8.1 2.a.ii	R
`dc:identifier`	A unique alphanumeric identifier (OID) included in any report instance generated using the template	This value shall be an Object identifier (OID) as specified in ITI TF-2x: Appendix B. A new value shall be assigned when elements outside of the **`head`** element are modified. It is permitted to retain the value when the **`head`** element is updated	R
`dc:type`	Indicates the type of XML document	Shall be "**`IMAGE_REPORT_TEMPLATE`**"	R
`dc:publisher`	The organizations who have published the template (e.g., RSNA, the local site)		R
`dc:rights`	Licensing considerations for the template		R
`dc:license`	A reference to a license that may govern the use of the template		R
`dc:date`	The date of unspecified purpose that could be the most recent modification of the template		R
`dc:creator`	An individual or group who primarily created this template		R
`dc:contributor`	An individual or group who contributed to the template		O
`dc:relation`	The identifier of a template which deprecates this template (i.e., a new version of this template)		O
`dc:language`	The language in which the template is written	ISO 639 two-letter language code: although template encoding is UTF-8 as specified in the <head> section, some special	O

Dublin Core Template Metadata Elements	Description	Vocabulary Constraint	Opt
		characters in different languages will need to use the unicode representation.	

The following additional metadata attributes may be associated with each report template in the `template_attributes` element:

Table 8.1.1-2: Other Metadata Elements for Report Templates

Template Metadata Elements	Description	Vocabulary Constraint	Opt
`top-level-flag`	Binary attribute that indicates when the template should not be a subsection of another template	xsd:boolean	O

`status`	Marks templates that should no longer be used for creating reports.	`DRAFT,`
		`ACTIVE,`
		`RETIRED`
			O

`user-list`	The users to which this template may apply, separated by commas		O
`provider-group-list`	The provider groups to which this template may apply, separated by commas		O

Report Creators may use these metadata elements to guide application behavior.

8.1.2 Section Attributes

Table 8.1.2-1 shows the attributes that may be associated with each section. These attributes are specified as custom data attributes in the HTML start tag for the section. Figure 8.1.2-1 shows an example.

Table 8.1.2-1: Attributes of Report Template Sections

HTML5 Attribute	Description	Vocabulary Constraint	Opt
`data-section-name`	A human readable name for the section		R
`data-section-required`	A flag indicating whether this section must appear in the report (can be deleted)	xsd:boolean	O
`id`	Linking identifier for coded content that		O

HTML5 Attribute	Description	Vocabulary Constraint	Opt
	corresponds to this item.		

```
<body>
   <section
      id="T002"
      data-section-name="Procedure:"
      >
      <header class="level1">Procedure:</header>
      <p>
         <!—Section content here-->
      </p>
   </section>
</body>
```

Figure 8.1.2-1: Example of `<section>` element

8.1.3 Report Template Fields

Many current reporting systems support the concept of fields, typically rendered as delimited by square brackets. Fields can serve many purposes, including:

- Emphasize a part of the report that frequently should be modified by the provider (e.g., left/right information). Default text may be provided, even where there is an expectation that it will be replaced or modified.
- Enable rapid navigation among parts of the report that are frequently modified (e.g., using rewind and fast-forward buttons to move between fields).
- Serve as a visual cue for more complex user interface behavior (e.g., verbal triggers, pick lists, and other field types).
- Represent a field that must be accepted or edited before the report can be finalized.

Fields shall be described only using the HTML **`select`** or **`input`** elements and attributes as shown in Table 8.1.3-1 and in the subsections that follow.

Table 8.1.3-1: Attributes of Report Template Fields

Field Type	HTML5 Element	HTML5 Attribute	HTML5 Attribute Value
Text	**input**	**type**	**text**, **textarea**
Numeric	**input**	**type**	**number**
Selection List	**select**	**multiple**	**single, multiple**
Date	**input**	**type**	**date**
Time	**input**	**type**	**time**

820 8.1.3.1 Field Attributes

Table 8.1.3.1-1 shows the attributes associated with fields of any type. These are specified as attributes of the HTML element for the field.

Table 8.1.3.1-1: Attributes of Report Template Fields

HTML5 Attribute	Description	Vocabulary Constraint	Opt
name	A human readable alphanumeric identifier for this field.		R
data-field-type	The nature of the information intended to be captured by this field	**TEXT, NUMBER, SELECT, DATE, TIME, MERGE**. See Sections 8.1.3.3 -8.1.3.8.	R
data-field-merge-flag	Indicates whether content from an outside source can be accepted as the value of this field. If this attribute is absent, the assumed value is 'N'.	xsd:boolean	O
data-field-verbal-trigger	A word or phrase to enable rapid navigation to the field with a voice command. If no verbal trigger is specified, the Field name serves as the default verbal trigger.		O
value	The value of the field if the user does not modify it. This value is shown when the report template is initially displayed.		O
data-field-completion-action	Indicates an action that may be taken if the field is not populated by the user of the Report Creator	**NONE** **ALERT** **PROHIBIT**	O
title	A hint that appears when the mouse hovers over the field		O
id	Linking identifier for coded content that corresponds to this item.		O

825 The values of the **data-field-completion-action** attribute shall be interpreted as follows:

- **NONE**: The user may modify the field in any way, including deletion, without restriction. If no value for the **data-field-completion-action** field is specified, **NONE** is the default.
- **ALERT**: If a field with this attribute value is blank or missing, the user is alerted at report 830 completion time. These fields may be deleted by the user, but such a deletion also will cause an alert at completion time.

- **PROHIBIT**: A value must be supplied for a field with this attribute value prior to report completion. Deletion is not allowed. If the field is blank, the report completion is prohibited.

8.1.3.2 Linkage Between Template Text and Template Fields

To signify that a field is semantically linked to specific template text, the text should be marked by a **label** element whose **for** attribute matches the **id** attribute of the associated field. If the text is modified, the **label** content associate with this field may be invalid. Figure 8.1.3.2-1 shows an example.

```
<label for="T010">The unenhanced liver attenuation is</label>
<input id="T010"
   name="unenhanced_liver"
   data-field-type="NUMERIC"
   type="number"
   title="Liver attenuation"
   data-field-signature-latitude="ALERT"
   value="0"
   max="1000"
   min="-1000"
   />
HU.
```

Figure 8.1.3.2-1: Example of Linkage Between Template Text and Template Field

Section 8.1.6.3 discusses possible application behavior when semantically linked text is modified by the user.

8.1.3.3 Text Field Attributes

Text fields may contain narrative text that is editable by the user. The main function of a text field is to delimit a part of the report for rapid navigation and possible modification. Text fields are currently in wide use by radiology speech recognition systems. Text fields shall be expressed using the HTML5 **input** element with **type** attribute = **text** (for single line text fields) or with **type** attribute = **textarea** (for multi-line text fields).

8.1.3.4 Numeric Field Attributes

Numeric fields contain a numeric value with associated optional range and optional units. Numeric fields shall be expressed using the HTML5 **input** element with **type** attribute = **number.** Table 8.1.3.4-1-3 shows the additional attributes associated with numeric fields.

Table 8.1.3.4-1: Attributes of Numeric Fields

HTML5 Attribute	Description	Vocabulary Constraint	Opt
`min`	The minimum value this field will accept, checked by the Report Creator		O
`max`	The minimum value this field will accept, checked by the Report Creator		O
`data-field-units`	The unit of measure for the number, such as "HU"	Unified Code for Units of Measure (UCUM)	O
`step`	Use step=1 for integers, or specify a step value smaller than 1 for real numbers. (e.g., 0.1, or 0.01, depending on desired precision).		O

8.1.3.5 Selection List Field Attributes

Selection list fields can take on a value selected by the user from a list of items. Each item on the list may be associated with text that should be displayed in the report if that item is selected. Selection lists may have a default value, which is displayed in the field when the template is applied. A user choice among the list items may be required or optional. A single selection may be required, or multiple selected elements may be allowed. Only the attributes and coded content associated with the item(s) selected by the user are associated with the report instance that is generated.

Selection list fields shall be expressed using the HTML5 **`select`** element. Table 8.1.3.5-4 shows the additional attributes associated with selection list fields.

Table 8.1.3.5-4: Attributes of Selection List Fields

HTML5 Attribute	Description	Vocabulary Constraint	Opt
`multiple`	`single`: Only one choice can be selected from the list of items, typically implemented as a menu or as radio buttons. `multiple`: More than one choice can be selected from the list of items, typically implemented as multiple check boxes or as a multi-select menu. Default if unspecified is `single`.	`single` `multiple`	O

8.1.3.5.1 Selection Items

Each selection item shall be expressed using the HTML5 `option` element within a `select` element. Table 8.1.3.5.1-1-5 shows the attributes associated with Selection Items.

885 **Table 8.1.3.5.1-1: Attributes of Selection Items**

HTML5 Tag	Description	Vocabulary Constraint	Opt
`name`	A human readable alphanumeric identifier of this item.		R
`value`	The text that should be displayed and incorporated into the report when this item is selected.	The value of this attribute must equal the content between the open and close tags	R
`data-template-UID`	The unique identifier of a template, as specified in its `dc:identifier`, that should be substituted at an arbitrary point in the document. [use `span` or `div` element to specify location]		O
`data-replacement-element-id`	The id of the element to be replaced with the selected content. [use `span` or `div` element with corresponding `id` tag]		C, required if `data-template-UID` is present
`label`	A comment or instruction on the meaning of this item.		O
`data-verbal-trigger`	A word or phrase that, when dictated, causes this item to be selected by the Report Creator		O
`selected`	Binary variable indicating whether this item should be used as the default value for this field.		O
`id`	Linking identifier for coded content that corresponds to this item.		O

If `data-template-UID` is specified, the template with that UID replaces the report text delimited by the element whose `id` corresponds to `data-replacement-element-id.` This function enables user selection of a menu item to trigger insertion of another template at an arbitrary point. 890

8.1.3.6 Date Field Attributes

Date fields accept calendar date information. Date fields shall be expressed using the HTML5 **`input`** element with **`type`** attribute = **`date`**.

8.1.3.7 Time Field Attributes

Time fields accept clock time information. Time fields shall be expressed using the HTML5 **`input`** element with **`type`** attribute = **`time`**.

8.1.3.8 Merge Field Attributes

Merge fields accept information from other sources, such as patient information from an HL7 order, measurements from an ultrasound device, or a region of interest (ROI) calculation from an imaging workstation. These fields can substantially increase the efficiency of the user and the accuracy of the report by reducing the need to re-dictate or re-enter these data. Table 8.1.3.8-1 shows the additional attributes associated with merge fields.

Table 8.1.3.8-1: Attributes of Merge Fields

HTML5 Attribute	Description	Vocabulary Constraint	Opt
`data-merge-identifier`	An opaque string label for an implementation specific value that is obtained by the Report Creator		C, required for Merge fields only

This profile does not define how data from other sources are incorporated into merge fields. Transforming the merge data to fit into the report is an undefined task for the Report Creator. Those functions depend on the capabilities of the Report Creator, and may or may not be present. Report Creators may Merge data into any field type, including text, numeric, and selection list fields.

8.1.4 Incorporating Templates into Other Templates

Templates may be incorporated into other templates, sometimes called "nested" or "modular" templates. The HTML5 **`embed`** element shall be used to indicate the location where a template should be included. The **`src`** attribute of the **`embed`** element shall contain a string concatenating the unique identifier for the template to be included, and "**`.html`**". The **`type`** attribute of the **`embed`** element shall be "**`text/html`**". Figure 8.1.4-1 shows an example of how content from Template2 could be embedded in Template1; Template1 will contain the HTML shown in the figure at the point where Template2 should be included:

```
   <embed src=”1.2.3.4.6.html” type=”text/html” />
```

Figure 8.1.4-1: Example of Incorporating Content from Template2.

When content from Template2 are embedded in Template1, identifiers shall be prefixed to avoid namespace conflicts.

8.1.5 Permitted HTML5 Formatting Tags

Table 8.1.5-1 shows the HTML5 tags that may be used to format and markup a template. Other formatting tags are permitted but will be ignored.

Table 8.1.5-1: Permitted HTML5 Formatting Tags

HTML5 Tag	Description
<a>	Hyperlink to a web resource specified by the **href** attribute
<br>	Line break
<em>	Emphasized text, often in italics
<img>	Image
<li>	List item
<ol>	Ordered list
<p>	Paragraph
<q>	Short quotation
<span>	Groups inline text and other elements
<strong>	Important text, often bold
<sub>	Subscripted text
<sup>	Superscripted text
<table>	Defines a table
<td>	Table cell
<th>	Header cell in a table
<tr>	Table row
<u>	Stylistically different text, often underlined
<ul>	Unordered list

8.1.6 Coded Content

Report components may be associated with coded content, such as terms from a controlled vocabulary that explicitly represent the semantics of the text. These references may be used to link text in the report template or the report instance with the associated machine-readable semantic content. These references enable applications to manage the relationship between report text and associated coded content during user editing.

8.1.6.1 Simple Format for Coded Content

The **head** element of each template contains three blocks of coded content:

1. Dublin Core attributes are expressed in **meta** tags.
2. The **template_attributes** block contains coded content that applies to the entire template.
3. The **coded_content** block contains coded content linked to specific elements in the **body** of the template.

Each **entry** element in the **coded_content** block shall be linked to an element in the **body** of the report when the **ORIGTXT** attribute of an **entry** matches the **id** attribute of an element in the **body**. The format for each **entry** shall contain at least one [1..*] **term** element, encoded in XML. Each term element shall contain exactly one **code_meaning** element, exactly one **code_value** element, and exactly one **coding_scheme_designator** element. **term** elements may be assigned a **type** attribute indicating the template attribute for which the term tuple is the value.

Table 8.1.6.1-1: Elements of a <term> Element

Element	Description	Vocabulary Constraint	Opt
code_meaning	Human readable text provided for the convenience of readers		R
code_value	A computer readable and computer searchable identifier that is unambiguous within the coding scheme denoted by the coding_scheme_designator		R
coding_scheme_designator	Uniquely identifies the resource where the code_value is linked to its code_meaning	An Object identifier (OID) as specified in ITI TF-2x: Appendix B.	R

Figure 8.1.6.1-1 shows a simple example in which the identifier **T002** is used to link a section in the **body** to a RadLex term specified in the **coded_content** block. By repeating the term block, a list of terms could be used to specify the coded content.

```
   <head>
      <!--other head items here-->
      <coded_content>
         <entry ORIGTXT="T002">
            <term>
               <code_meaning>Procedure</code_meaning>
               <code_value>RID1559</code_value>

   <coding_scheme_designator>2.16.840.1.113883.6.256</coding_scheme_designato
r>
            </term>
         </entry>
         <!—additional entries here-->
      </coded_content>
   </head>
   <body>
      <section
         id="T002"
         class="level1"
         data-section-name="Procedure:">
         <header class="level1">Procedure:</header>
         <p>
         <!—Section content here-->
         </p>
      </section>
   <!--remainder of template body here-->
   </body>
```

Figure 8.1.6.1-1: Simple Coded Content Example

8.1.6.2 Complex Coded Content

Unfortunately, an unstructured list of terms is sometimes insufficient to express even simple concepts. Listed below are examples of coded content too complex to express in an unstructured list:

1. Concept: “Pain in the left upper quadrant and the right lower quadrant” Term list: (PAIN, RLQ, LUQ). Does the pain apply to both RLQ and RUQ, or just one?
2. Concept: “Numbness and pain the left face and right shoulder”. Term list: (LEFT, FACE, RIGHT, SHOULDER, PAIN, NUMBNESS). Which body parts are specified, and which symptom belongs with which part?
3. Concept: “A mass projects from the medial aspect of a calcified left kidney”. Term list: (MASS, PROJECTS, MEDIAL, CALCIFIED, LEFT, KIDNEY). Is the kidney or the mass calcified?

The MRRT white paper proposes a simple recursive structure, called a Term Set, to express more complex coded content. However, the use of HL7 CDA encoding or other standard methods may also provide sufficiently rich expressions that could be passed along from a report template to a report instance. This profile does not address expressions of coded content other than lists of terms.

8.1.6.3 Managing Coded Content during Template Editing

Report creation tools typically provide a text editor metaphor, allowing the user to select, delete, edit, or add text at an insertion point. When text linked to coded content is edited by the user, the coded entries underlying the text may no longer be valid. In the worst case, the user might edit the text to negate the sentence. How these dependencies are managed will be left to each system implementation. Possible actions include doing nothing, warning the user, or deleting associated coded content.

8.1.7 Merging Data into Report Templates

It may be desirable to merge data into a template field from another source. There are many possible sources from which this merge data might be derived:

- The HL7 order for this study
- Other information in the radiology information system (RIS)
- The DICOM image header for this study
- A DICOM structured report for this study
- Previous reports for this same patient
- The electronic medical record for this same patient

The MRRT white paper proposes a Merge Retrieval Method that retrieves these data from a Merge Source. However, a simpler and more prevalent method for the retrieval of merge data is described in this profile. A single opaque identifier, which is an attribute of each Merge field, may be used by an application to retrieve data to merge into the template. How an application manages the correspondence between this identifier and specific merge capabilities is left to each system implementation.

8.1.8 Relationships among Templates

It may be useful for applications to maintain relationships among templates. However, this profile specifies that `dc:relation` attribute shall be used only to represent the “deprecated by” relationship.

Other relationships, such as creating a “CT Abdomen” template, with children “CT Abdomen with IV contrast” and “CT Abdomen without, then with, IV contrast”, with children inheriting some attributes and features of the parent template are beyond the scope of this profile.

8.2 Report Template Query Response Structure

The format for report templates is expressed in XML.

Each template shall support the constraints described below. The template:

1. Shall begin with exactly one [1..1] XML declaration: **`<?xml version="1.0" encoding="UTF-8"?>`** declaring the character set used.
2. Shall contain exactly one [1..1] **`templates`** element.
    a. The `templates` element may contain [0..*] `template` elements, one for each matching template.
        i. The `template` element shall contain an `href` attribute to indicate a URL to retrieve the template.
        ii. The `template` element shall contain exactly one [1..1] **`meta`** element declaring the character set used: **`<meta charset="UTF-8">`**.
        iii. The `template` element shall contain exactly one [1..1] **`title`** element containing the name of the template.
        iv. The `template` element may contain exactly one [1..1] **`style`** element with internal CSS style elements.
        v. The `template` element shall comply with all other HTML5 constraints.

```
<?xml version=“1.0” encoding=“UTF-8”?>
<templates>
   <template href=”http://<location>/IHETemplateService/1.2.3.4.5”>
      <meta name=”DC.title” content=”CT Brain” />
      <meta name=”DC.identifier” content=”1.2.3.4.5” />
      <meta name=”DC.type” content=”Image report template” />
      <meta name=”DC.language” content=”en” />
      <meta name=”DC.publisher” content=”Radiological Society of North
America (RSNA)” />
      <meta name=”DC.rights” content=”May be used freely, subject to license
agreement” />
      <meta name=”DC.license” content=”http://www.radreport.org/license.pdf”
/>
      <meta name=”DC.date” content=”2012-03-28” />
      <meta name=”DC.creator” content=”Flanders AE, et al.” />
      <meta name=”DC.contributor” content=”Bozkurt S [coder]” />
      <meta name=”DC.contributor” content=”Kahn CE Jr [editor]” />
      <meta name=”DC.contributor” content=”American Society of Neuroradiology
(ASNR)” />
      <meta name=”status” content=”ACTIVE” />
      <meta name=” top-level-flag” content=”0” />
      <template_attributes>
         <top_level_flag>true</top_level_flag>
         <term type="modality">
            <code_meaning>computed tomography</code_meaning>
            <code_value>RID10321</code_value>

   <coding_scheme_designator>2.16.840.1.113883.6.256</coding_scheme_designato
r>
         </term>
         <term type="body part">
            <code_meaning>brain</code_meaning>
            <code_value>RID6434</code_value>

   <coding_scheme_designator>2.16.840.1.113883.6.256</coding_scheme_designato
r>
         </term>
      </template_attributes>
   </template>

   <!--Additional template elements may appear here-->

</templates>
```

Figure 8.2-1: Example of the Report Template Query Response

Relevant `meta` tag name values can be found in 8.1.