Copyright Information

All Licensed Software is protected by the copyright laws of the United States and by applicable international treaties. No rights under copyrights are transferred to Customer, except as specifically provided in the License Grant.

Trademark Information

GE and the GE Monogram are trademarks or registered trademarks of the General Electric Company. All other product names and logos are trademarks or registered trademarks of their respective owners.

Confidentiality and Proprietary Rights

This document is the confidential property of GE and/or its affiliated entities. It is furnished to, and may only be used by entities that have entered into an appropriate non-disclosure agreement (“NDA”) with GE and may only be used in accordance with the terms of that agreement. If you are a customer of GE Healthcare IT, in addition to any NDA you may be required to execute additional license and service terms which are applicable specifically to the use and support of the application programming interfaces (API's) further described in this documentation.

The access and use of this document is restricted to customers and their employees. The user of this document agrees to protect the confidentiality of the information contained herein and GE’s proprietary rights as expressed herein and not permit access to this document by any person or entity who has not entered into an NDA with GE. In no case may this document or any portion hereof be accessed, made available, examined, or copied for the purpose of developing, marketing, or supporting any system or computer program similar to the GE software. No part of this document may be copied without the prior written permission of GE. The information in this document is subject to change by GE without notice.

Inquiries regarding permission for use of material contained in this document should be addressed to:
GE Healthcare ATTN: General Counsel, 500 W. Monroe, Chicago, IL. 60661 USA

Publication Date and Document Numbers

May 7, 2018
DOC2073977
2107663-001

Service Pack

SP9
# Table of Contents

1 About This Manual ........................................................................................................ 4  
   1.1 Introduction ........................................................................................................... 4  
   1.2 Purpose ................................................................................................................ 4  
1.3 Applications Support Using iCenter, eService, and Answerline ............................... 4  
   1.4 Contacting the ROC ................................................................................................ 5  
   1.5 Acronyms and Abbreviations ............................................................................... 5  
2 Create a ZFP URL ......................................................................................................... 7  
3 ZFP Inbound Launch .................................................................................................... 11  
   3.1 Introduction ........................................................................................................... 11  
   3.2 ZFP URL Launch ................................................................................................... 12  
      3.2.1 ZFP Non-Encrypted Static URL .................................................................. 12  
      3.2.2 Patient Timeline Non-Encrypted Static URL ................................................. 13  
      3.2.3 ZFP Dynamically Encrypted URL Launch ................................................... 14  
      3.2.4 URL Parameters ......................................................................................... 14  
      3.2.5 ZFP URL Launch for MultiSite Command .................................................... 22  
      3.2.6 ZFP Launch Configurations (Information for Calling Application) ............... 23  
   3.3 ZFP Java Script Launch .......................................................................................... 24  
   3.4 Encryption of URL Parameters for ZFP Launch from External Applications ......... 27  
   3.5 Decryption for Payload ......................................................................................... 27  
   3.6 Generating Hexadecimal Key for Open SSL Encryption ......................................... 28  
   3.7 ZFP Launch Configurations (Information for GE Service Engineer) ........ ........... 28  
   3.8 ZFP Time-out in URL Launch Mode ...................................................................... 29  
   3.9 ZFP Outbound Launch .......................................................................................... 29  
Appendix A: Authentication .......................................................................................... 30  
Appendix B: Open Desktop Integration with Microsoft IE 8 and 9 ................................. 31  
Appendix C: ZFPApiTool ................................................................................................ 33  
Appendix D: Creating a ZFP URL .................................................................................. 37  
Appendix E: Using Cryptography.dll to Encrypt URL ................................................... 39  
Appendix F: CWeb to ZFP Migration .............................................................................. 41  
Appendix G: Proxy Launch from Non-Web-based Application ....................................... 44
Appendix H: Proxy Launch with Single ZFP (ZFP Configured Against EA and CPACS as DICOM Sources).......................................................................................................................................................... 46
Appendix I: General Troubleshooting ........................................................................................................................................................................ 47
Chapter 1: About This Manual

1 About This Manual

**Prerequisite:** This manual is for developers with significant C++, Java, and XML experience. If you do not meet this prerequisite, do not attempt to use the API. Contact your GE service representative for assistance.

1.1 Introduction

This manual contains detailed information for third-party applications to integrate with the Universal Viewer. It contains:

- Centricity Universal Viewer Zero Footprint client (ZFP) integrations specific to how to integrate a client with ZFP via a ZFP URL interface and ZFP JavaScript Interface.

1.2 Purpose

This manual is targeted for advanced users who will implement and test third-party applications integrating with the Universal Viewer.

1.3 Applications Support Using iCenter, eService, and Answerline

To request applications support, English-speaking customers may open a support request in GE Healthcare iCenter™ or eService. Other customers should contact their Remote Online Center support personnel.

With GE Healthcare iCenter you can:

- Initiate requests for service and applications support from the web.
- Get rapid, online access to the Remote Online Center (ROC) where expert service engineers review and respond to service requests quickly.
- View status of open service requests.
- View service history and reports about your systems, including uptime, remote fix and service call trends, and other service delivery metrics.

If you do not have access to iCenter or eService, contact your Director of Service or the Remote Online Center to open an account. English-speaking customers in the United States and Canada can also use the following resources:

- 1 (855) 762-6650
- 1 (800) 437-1171, Option 3
### 1.4 Contacting the ROC

Use the following information to contact the ROC (Remote Online Center).

<table>
<thead>
<tr>
<th>OLC/US (US and Canada)</th>
<th>OLC/Europe</th>
<th>OLC/ANZ</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-855-6650</td>
<td>+33 (0) 1 30 831300</td>
<td>61-2-316-3700</td>
</tr>
</tbody>
</table>

### 1.5 Acronyms and Abbreviations

<table>
<thead>
<tr>
<th>Term</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>AD</td>
<td>Active Directory</td>
</tr>
<tr>
<td>API</td>
<td>Application Programming Interface</td>
</tr>
<tr>
<td>ASR</td>
<td>Automatic Speech Recognition</td>
</tr>
<tr>
<td>CJK</td>
<td>Chinese, Japanese, and Korean</td>
</tr>
<tr>
<td>CTR</td>
<td>Cardio-Thoracic Ratio</td>
</tr>
<tr>
<td>CDP</td>
<td>Conference Display Protocol</td>
</tr>
<tr>
<td>DDP</td>
<td>Default Display Protocol</td>
</tr>
<tr>
<td>DICOM</td>
<td>Digital Imaging and Communication</td>
</tr>
<tr>
<td>FTP</td>
<td>File Transfer Protocol</td>
</tr>
<tr>
<td>GSPS</td>
<td>Grey Scale Presentation State</td>
</tr>
<tr>
<td>HP</td>
<td>Hanging Protocol</td>
</tr>
<tr>
<td>HL7</td>
<td>Health Level Seven</td>
</tr>
<tr>
<td>IDI</td>
<td>Independent Mammography Viewer</td>
</tr>
<tr>
<td>IDR</td>
<td>Integrated Desktop Reporting</td>
</tr>
<tr>
<td>IVAPI</td>
<td>Universal Viewer web client Application Programming Interface .NET component interface</td>
</tr>
<tr>
<td>IHE</td>
<td>Integrating the Healthcare Enterprise initiative; also the set of technical frameworks established by it.</td>
</tr>
<tr>
<td>MIP</td>
<td>Maximum Intensity Projection</td>
</tr>
<tr>
<td>MPR</td>
<td>Multi-Planar Reconstruction</td>
</tr>
<tr>
<td>NIO</td>
<td>Non-Image Object</td>
</tr>
<tr>
<td>PACS</td>
<td>Picture Archive and Communication System</td>
</tr>
<tr>
<td>RF</td>
<td>Radio Frequency</td>
</tr>
<tr>
<td>RIS</td>
<td>Radiology Information System</td>
</tr>
<tr>
<td>ROI</td>
<td>Region of Interest</td>
</tr>
<tr>
<td>RP</td>
<td>Requested Procedure</td>
</tr>
<tr>
<td>-----</td>
<td>--------------------------------------</td>
</tr>
<tr>
<td>TCD</td>
<td>Transverse Chest Diameter</td>
</tr>
<tr>
<td>THD</td>
<td>Transverse Heart Diameter</td>
</tr>
<tr>
<td>UID</td>
<td>Unique Identifier as defined in the DICOM standard</td>
</tr>
<tr>
<td>URL</td>
<td>Uniform Resource Locator</td>
</tr>
</tbody>
</table>
Chapter 2: Create a ZFP URL

2 Create a ZFP URL

Answer the questions below to help create the ZFP URL. The URL will integrate the ZFP DICOM Viewer when it is launched by a third party, such as an EMR.

1. Will you be launching ZFP from an EMR application/Hospital SSO portal/another third-party application?
   Yes - Go to #2
   No - Use ZFP standalone URL: http://<ZFPSERVERIP>/ZFP

2. Do you want users to be logged in automatically while using ZFP in “standalone mode” (that is, ZFP is not opened with a specific patient or study, and users can access predefined worklists and use the Search functionality)?
   Yes – Use ZFP standalone URL with user credentials: http://<ZFPSERVERIP>/ZFP#un=<UserName>&pw=<Password>
   No – Go to #3

3. Do you need to display the title bar on ZFP?
   Yes - http://<ZFPSERVERIP>/ZFP?mode=proxy&titlebar=on
   No - http://<ZFPSERVERIP>/ZFP?mode=proxy&titlebar=off

4. Do you need to display ZFP in lights off mode?
   Yes - http://<ZFPSERVERIP>/ZFP?mode=proxy&lights=off
   No - http://<ZFPSERVERIP>/ZFP?mode=proxy&lights=on

5. Do you need your users to be able to use the Search functionality upon launching ZFP?
   Yes - Enable the following configuration in Appsettings.config: 'EnablePatSearchFieldsOnInboundLaunch=true'
   No - Disable the following configuration in Appsettings.config: 'EnablePatSearchFieldsOnInboundLaunch=false'

6. Do you need to display prior studies in the Navigator and the Series Selector upon launching ZFP?
   Yes - Enable the following configuration in Appsettings.config: 'AccessToPriors=true'
   No - Disable the following configuration in Appsettings.config: 'AccessToPriors=false'

7. When do you need to display the worklist upon launching ZFP from EMR application?
   Always - http://<ZFPSERVERIP>/ZFP?mode=proxy&lights=off&titlebar=on#Showlist
   Never - http://<ZFPSERVERIP>/ZFP?mode=proxy&lights=off&titlebar=on#ViewAll
   Only when single study is available - http://<ZFPSERVERIP>/ZFP?mode=proxy&lights=off&titlebar=on#View

8. With a CPACS back end, what information is available on the EMR to launch ZFP viewer?
   Patient ID - http://<ZFPSERVERIP>/ZFP?mode=proxy&lights=off&titlebar=on#View&ris_pat_id=<Patient_ID>
Chapter 2: Create a ZFP URL

9. With a Centricity EA back end, what information is available on the EMR to launch ZFP viewer?
   - Patient ID - http://<ZFPSERVERIP>/ZFP?mode=proxy&lights=off&titlebar=on&pid=<Patient_ID>
   - Patient ID + Accession Number - http://<ZFPSERVERIP>/ZFP?mode=proxy&lights=off&titlebar=on&pid=<Patient_ID>&san=<Accession_Number>
   - Study Instance UID - http://<ZFPSERVERIP>/ZFP?mode=proxy&lights=off&titlebar=on#View&sui=<SUID>

10. Will your site require users to enter their credentials upon launching ZFP every time from the EMR/third-party application?
    - Yes - http://<ZFPSERVERIP>/ZFP?mode=proxy&lights=off&titlebar=on#View&sui=<SUID>&un=<UserName>&pw=<Password>
    - No - http://<ZFPSERVERIP>/ZFP?mode=proxy&lights=off&titlebar=on#View&sui=<SUID>

For ZFP launch from the GE Health Cloud or Centricity 360:
http://<ZFPSERVERIP>/ZFP?mode=proxy&lights=off&titlebar=on#View&sui=<SUID>&caseld=<CASEID>&AuthKey=<AUTHKEY>

Note: If you are using Windows Authentication to authenticate the user with Microsoft Internet Explorer or Chrome, then the browser may remember the user credentials so the user does not have to authenticate every time ZFP is launched. With other supported browsers, the end user will be required to enter their credentials every time ZFP is launched.

11. Does ZFP need to connect to multiple domains? For example, with a Centricity EA back end, do you have PIX enabled? Or with a CPACS back end, do you have multiple Patient ID domains?
    - Yes - Add 'authority' parameter to the end of URL as follows:
      http://<ZFPSERVERIP>/ZFP?mode=proxy&lights=off&titlebar=on#View&sui=<SUID>&authority=<Domain_ID>
    - No - Do not add an 'authority' parameter to the end of the URL

Note: With a Centricity PACS back end, use the short code for Assigning_Authority from the IMS database. With an EA back end, use the domain from the DBConnections.config file in ZFP as follows:
If the domain specified in the DBConnections.config file in ZFP is:
   Domain="SYS_A&amp;1.2.3.4.5.1000&amp;ISO", then use authority=SYS_A.

12. Should the search controls be visible?
    - Yes – http://localhost/ZFP?searchControl=on
    - No – http://localhost/ZFP?searchControl=off
Chapter 2: Create a ZFP URL

Note: search controls can also be shown / hidden through configuration (check “Display search filter on proxy launch” in “Study Search” tab of the Admin Desktop).

13. Does the EMR/third party application launching ZFP support encryption?
   
   Yes - Refer Sections 3.2.2 Patient Timeline Non-Encrypted Static URL and 3.2.3 ZFP Dynamically Encrypted URL
   
   No - Use the URL generated in steps above (#8, 9, 10)

14. Does the EMR/third party application launching ZFP use JavaScript?
   
   Yes - Refer Section 3.3 ZFP Java Script Launch
   
   No - Use the URL generated in steps above (#8, 9, 10)

15. Would you get static URL from CCG or any RIS system?
   
   Yes - Request the static URL from GE Systems Integration Engineer
   
   No - Use the URL generated in steps above (#8, 9, 10)

16. Will the static URL from CCG or RIS system be encrypted?
   
   Yes - Request the static URL from GE Systems Integration Engineer
   
   No - Use the URL generated in steps above (#8, 9, 10)

17. Does the EMR have static URL stored for studies to be launched in ZFP?
   
   Yes - Refer Appendix F: CWeb to ZFP Migration
   
   No - Use the URL generated in steps above (#8, 9, 10)

If you need to launch ZFP using additional parameters supported by Centricity EA or Centricity PACS back ends, refer to Section 3.2.4 URL Parameters for ‘Query Parameters supported by EA or CPACS back ends’ and append them to the end of the URL.

The diagrams below describe how ZFP is positioned within the hospital infrastructure in terms of the EMR, PACS, EA, and so on. There are two configurations. The first shows ZFP in an environment where the CCG Broker updates records in the EMR. The second shows ZFP in an environment where no CCG Broker is required.
3 ZFP Inbound Launch

3.1 Introduction

Centricity Universal Viewer Zero Footprint client (ZFP) supports two modes of integration when launched from external applications:

- **URL Launch**
  Used for majority of integrations by EMRs for integration purposes.

- **Java Script Launch**
  This is available only in non-XDS environments with DICOM viewing capabilities. It is used when tighter context requirements are required.

Summary of differences are:

<table>
<thead>
<tr>
<th></th>
<th>URL launch</th>
<th>Java Script launch</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reuse of same web browser instance to change context</td>
<td>Via web browser e.g., on IE8, 9, 10: Tools -&gt; Internet Options -&gt; General Tab -&gt; Tabs -&gt; Settings -&gt; Open links from other programs in : Select 'The current Tab or Window'</td>
<td>Programmatically (using WebViewerProxy.js or WebViewerProxy_IE8.js for IE8 browser)</td>
</tr>
<tr>
<td>Initialization and reuse of web browser and ZFP prior opening of a study</td>
<td>Not available</td>
<td>Yes</td>
</tr>
<tr>
<td>Tight integration with third party</td>
<td>Not available</td>
<td>Available through IsConnected() function interface in WebViewerProxy.js or WebViewerProxy_IE8.js for IE8 browser</td>
</tr>
<tr>
<td>Patient Timeline in XDS Environment</td>
<td>Available</td>
<td>Not Available</td>
</tr>
</tbody>
</table>

**Authentication**

Typical integrations use anonymous user authentication to enable single sign on-like capabilities, that is, users are not synchronized between ZFP and the calling application. The calling application uses an encrypted authentication key (password) to authenticate the request.

The table below summarizes the various authentication options:
### 3.2 ZFP URL Launch

#### 3.2.1 ZFP Non-Encrypted Static URL

ZFP non-encrypted static token URL comprises of 3 parts:

```
Base URL                      Query_parameters
https://<ZFPServerIP>/ZFP?mode=proxy#view<&query_parameters>&
```

**Authentication and Audit Parameters**

\[\text{un}=<\text{UserName}>\&\text{pw}=<\text{Encrypted_Password}>\&\text{custom}=<\text{CustomString}>\]

**Base URL**

Base URL consists of:
- https – secure protocol.
- `<ZFPServerIP>` - Host name or IP address of the ZFP server (custom to site).
- `zfp?mode=proxy#view&` – This command indicates that an external application is launching ZFP.

**Query_parameters**

This can be a single or combination of query parameters explained in the section ‘URL Parameters’ below. For example: `pid=<patientID>&authority=<domainID>&`

**Authentication and Audit Parameters**

Authentication and Audit Parameters consists of:
- \text{un} = User Name . (anonymous login username e.g., “ZFP User”)
• pw = Encrypted password or token
• custom – This is a custom string that will be logged in Windows log and ATNA repository (if available).
• custom='audit user name’
• E.g., un=ZFPUser&pw=fger%2typosu&custom=audit_user_name

Thus, the static URL built above looks like this:

http://<ZFPServerIP>/ZFP?lights=on&mode=proxy#view&pid=<patientID>&authority=<domainID>&un=ZFPUser&p
w=fger%2typosu& custom='audit_user_name’

Note: You can also launch ZFP DICOM viewer in “standalone mode” (that is, not opened with a specific patient or
study, and users can access predefined worklists and use the Search functionality), while avoiding users to
authenticate. This can be done using a URL as follows:

http://<ZFPSERVERIP>/ZFP#un=<UserName>&pw=<Password>

3.2.2 Patient Timeline Non-Encrypted Static URL

Base URL

https://<ZFPServerIP>/ZFP/PatientTimeline?

Query_parameters

PatientId=<Patient-id>&DomainId=<DomainId>&

Authentication and Audit Parameters

un=<UserName>&pw=<Encrypted_Password>

Base URL

Base URL consists of:

• https – secure protocol.
• <ZFPServerIP> - Host name or IP address of the ZFP server (custom to site).
• zfp/PatientTimeline? – This command indicates that an external application is launching ZFP.

Query_parameters

This can be a single or combination of query parameters explained in section ‘URL Parameters’ below. For example
PatientId=<patientID>&DomainId=< domainID> &
Chapter 3: ZFP Inbound Launch

Authentication and Audit Parameters

Authentication and Audit Parameters consists of:

- **un**: User Name (anonymous login username e.g., “ZFP User”)
- **pw**: Encrypted password

  ```
  https://<ZFPServerIP>/ZFP/PatientTimeline?PatientId=<patientID>&DomainId=<domainID>&un=ZFP User&pw=fger%2typosu
  ```

3.2.3 ZFP Dynamically Encrypted URL Launch

ZFP lets you securely integrate from the calling application by providing an option for dynamic encryption of the URL query parameters, authentication and audit parameters.

If customer site decides to use encrypted URL, then the URL would look like this:

```
http://<ZFPPIPServer>/ZFP?mode=proxy#pl=<Encrypted_Payload>
```

For Patient Timeline in an XDS environment, then the URL would look like this:

```
http://<ZFPPIPServer>/ZFP/PatientTimeline?pl=<Encrypted_Payload>
```

The encrypted payload is the static token URL as described above (Figure: ZFP URL Launch) with a UTC timestamp value appended to the username parameter with a pipe symbol e.g., ... | 20151213134546. This timestamp checks the validity of the URL, that is, the URL expires after a configured time.

Third-party integration vendors need ZFP.Cryptography.dll (explained in Appendix E) or to use AES algorithm on their own to programmatically generate the URL encryption and construct the payload. The steps to generate an encrypted payload are described in Appendix D: Creating a ZFP URL.

3.2.4 URL Parameters

1) **Base**: https://<zfpserver>/ZFP?mode=proxy&view&

   Here <zfpserver> is specified at the customer site.

   **Base**: https://<ZFP ServerIP>/ZFP/PatientTimeline?

   This is the base URL for Patient Timeline in an XDS environment.

2-a) **Optional** query parameters with ZFP DICOM Viewing that control the appearance of the viewer must appear before the hash symbol (#).

- **Title bar**
  - titlebar=<On/Off>
  - Example: http://<ZFP ServerIP>/ZFP?titlebar=off&mode=proxy#view...

- **Lights**
  - lights=on/off
  - Example: http://<ZFP ServerIP>/ZFP?lights=on&mode=proxy#view...

   **Note**: These parameters are not available with Patient Timeline in XDS environments.
2-b) **Optional** URL parameters pass the patient information to the Patient Timeline application and modify the appearance of the user interface for the ZFP-XDS Enabled viewer. The table below shows the URL parameters applicable to launching the Zero Foot Print -XDS Enabled application. The URL parameters are added to the end of the ZFP-XDS Enabled client, separated by an ampersand (&).

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FirstName</td>
<td>First Name</td>
<td>Specify first name. The value is case-insensitive.</td>
</tr>
<tr>
<td>LastName</td>
<td>Last Name</td>
<td>Specify last name. The value is case-insensitive.</td>
</tr>
<tr>
<td>DateOfBirth</td>
<td>Date of Birth</td>
<td>Specify date of birth in mm/dd/yyyy format.</td>
</tr>
<tr>
<td>Gender</td>
<td>M or F or U</td>
<td>Specify the gender: M - Male, F - Female, and U - Unknown.</td>
</tr>
<tr>
<td>BannerMode*</td>
<td>C</td>
<td>Display patient banner in Compact (List) mode.</td>
</tr>
<tr>
<td>ViewMode*</td>
<td>C or M</td>
<td>Specify view mode for document lists. C - List (Compact) View, and M - Matrix View.</td>
</tr>
<tr>
<td>avL</td>
<td>ALL</td>
<td>Automatically load the latest document for the selected patient when opening a patient record (either through URL or from the Patient Search).</td>
</tr>
</tbody>
</table>

* Patient Timeline uses the setting in the configuration file(s) for BannerMode and ViewMode if the parameter or its value is not specified in the URL. Refer to the chapter on Configuring ZFP XDS Enabled in the Centricity Universal Viewer Zero Footprint Client Installation and Upgrade Manual.

Example: https://3.232.166.238/ZFP/PatientTimeline?PatientId=2222&DomainId=1.2.840.113619.20.2.6.1.3.36.121.110&BannerMode=C&ViewMode=C opens the Patient Timeline application with a Compact Banner, and in Compact View.

2-c) Other query parameters that contain patient and or study identifiers must appear after the hash. These are different based on the back end (for legacy reasons) but can be used to query any of the configured backend (EA, CPACS or both). Please refer section Appendix H: Proxy Launch with Single ZFP (ZFP Configured Against EA and CPACS as DICOM Sources) for more details. These sections are described in two tables in this section. Parameters that appear here can be encrypted by the calling application as described in the section “Encryption of URL Parameters for ZFP Launch from External Applications.”

<table>
<thead>
<tr>
<th>Query parameters on Centricity Enterprise Archive</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Patient level launch</td>
<td></td>
</tr>
<tr>
<td>pid=&lt;patientID&gt;</td>
<td>Not supported with patient timeline in XDS environments.</td>
</tr>
<tr>
<td>Patient identifier + authority</td>
<td>Supported with patient timeline in XDS environments.</td>
</tr>
<tr>
<td>pid=&lt;patientID&gt;&amp;authority=&lt;domainID&gt;</td>
<td></td>
</tr>
<tr>
<td>For environments with multiple patient identifiers.</td>
<td></td>
</tr>
<tr>
<td>Query parameters on Centricity Enterprise Archive</td>
<td>Notes</td>
</tr>
<tr>
<td>-------------------------------------------------------------------------------------------------------------------</td>
<td>----------------------------------------------------------------------</td>
</tr>
<tr>
<td>Patient identifier + Domain Id</td>
<td>Supported with patient timeline in XDS environments.</td>
</tr>
<tr>
<td>PatientId=&lt;patientId&gt;&amp;DomainId=&lt;domainId&gt;</td>
<td></td>
</tr>
<tr>
<td><strong>Patient identifier + Domain Id + HealthcareFacilityTypeCodes + HealthcareFacilityTypeCodeSystemName</strong></td>
<td>Supported with patient timeline in XDS environments.</td>
</tr>
<tr>
<td>PatientId=&lt;patientId&gt;&amp;DomainId=&lt;domainId&gt;&amp;hcftc=&lt;healthcareFacilityTypeCode&gt;^&lt;healthcareFacilityTypeCodeSystemName&gt;</td>
<td></td>
</tr>
<tr>
<td>Format: hcftc=value1^codeSystemName1~value2^codeSystemName2</td>
<td></td>
</tr>
<tr>
<td>*It can have multiple hcftc values. Separator should be “~”</td>
<td></td>
</tr>
<tr>
<td>*healthcareFacilityTypeCode values (value1, value2...etc) serves as identifier and basing on iti18 standards, are case sensitive.</td>
<td></td>
</tr>
<tr>
<td>Example: hcftc=123^Facility%20Code<del>456^Facility%20Code</del>789^Hospital%20Code</td>
<td></td>
</tr>
<tr>
<td>The example above displays all documents with HealthcareFacilityTypeCode “123” with CodeSystemName “Facility Code”. It also displays “456” with CodeSystemName “Facility Code” and “789” with CodeSystemName “Hospital Code”.</td>
<td></td>
</tr>
<tr>
<td>*It can also be used without the CodeSystemName by not adding “^” after the hcftc value. Note however that this will match all documents with the same hcftc values.</td>
<td></td>
</tr>
<tr>
<td>Example: hcftc=123~456</td>
<td></td>
</tr>
<tr>
<td>The example above displays all documents with HealthcareFacilityTypeCode “123” and “456” regardless of CodeSystemName.</td>
<td></td>
</tr>
<tr>
<td><strong>Patient identifier + Domain Id + PracticeSettingCodes + PracticeSettingCodeSystemName</strong></td>
<td>Supported with patient timeline in XDS environments.</td>
</tr>
<tr>
<td>PatientId=&lt;patientId&gt;&amp;DomainId=&lt;domainId&gt;&amp;psc=&lt;practiceSettingCode&gt;^&lt;practiceSettingCodeSystemName&gt;</td>
<td></td>
</tr>
<tr>
<td>Format: psc=value1^codeSystemName1~value2^codeSystemName2</td>
<td></td>
</tr>
<tr>
<td>*Can have multiple psc values. Separator should be “~”</td>
<td></td>
</tr>
<tr>
<td>*practiceSettingCode values (value1, value2...etc) serves as identifier and basing on iti18 standards, are case sensitive.</td>
<td></td>
</tr>
<tr>
<td>Example: psc=123^Practice%20Code<del>456^Practice%20Setting</del>789^Practice%20Code</td>
<td></td>
</tr>
<tr>
<td>The example above has the same effect as hctfc example, but applied on PracticeSettingCode instead.</td>
<td></td>
</tr>
<tr>
<td>*Can also be used without the CodeSystemName by not adding “^” after the psc value. Note however that this will match all documents with the same psc values.</td>
<td></td>
</tr>
<tr>
<td>Example: psc=123~456</td>
<td></td>
</tr>
<tr>
<td>Above example has same effect as hctfc example above but applied on PracticeSettingCode instead.</td>
<td></td>
</tr>
</tbody>
</table>
### Query parameters on Centricity Enterprise Archive

<table>
<thead>
<tr>
<th>Format</th>
<th>Logic</th>
</tr>
</thead>
<tbody>
<tr>
<td>Patient identifier + Domain Id + HealthcareFacilityTypeCodes + HealthcareFacilityTypeCodeSystemName + PracticeSettingCodes + PracticeSettingCodeSystemName</td>
<td>(hcftc=A AND psc=X) OR (hcftc=A AND psc=Y)</td>
</tr>
<tr>
<td>PatientId=&lt;patientId&gt;&amp;DomainId=&lt;domainId&gt;&amp;hcftc=&lt;healthcareFacilityTypeCode&gt;&amp;hcftcSystemName=&lt;healthcareFacilityTypeCodeSystemName&gt;&amp;psc=&lt;practiceSettingCode&gt;&amp;pscSystemName=&lt;practiceSettingCodeSystemName&gt;</td>
<td>(hcftc=A OR hcftc=C) AND (psc=X OR psc=Y)</td>
</tr>
<tr>
<td>Supports pairing of the two XDS metadata HealthcareFacilityTypeCode (hcftc) and PracticeSettingCode (psc) to filter out specifically documents.</td>
<td></td>
</tr>
<tr>
<td>Example: hcftc=123^Facility&amp;psc=ABC^PracCode&amp;hcftc=456^Facility&amp;psc=BCD^PracCode</td>
<td>Above example displays all the documents with HealthcareFacilityTypeCode “123” with CodeSystemName “Facility” and PracticeSettingCode “ABC” with CodeSystemName “PracCode”. Same with the second pair; it will also display all the documents with HealthcareFacilityTypeCode “456” with CodeSystemName “Facility” and PracticeSettingCode “BCD” with CodeSystemName “PracCode”.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Study level launch</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>accession number only</td>
<td>Not supported with patient timeline in XDS environments.</td>
</tr>
<tr>
<td>san=&lt;acc&gt;</td>
<td></td>
</tr>
<tr>
<td>accession number+Pat id</td>
<td>Not supported with patient timeline in XDS environments.</td>
</tr>
<tr>
<td>san=&lt;acc&gt;&amp;pid=&lt;patientID&gt;</td>
<td></td>
</tr>
<tr>
<td>Used where accession # is not unique.</td>
<td></td>
</tr>
<tr>
<td>accession number+Pat id + authority</td>
<td>Supported with patient timeline in XDS environments.</td>
</tr>
<tr>
<td>san=&lt;acc&gt;&amp;pid=&lt;patientID&gt;&amp;authority=&lt; domainID&gt;</td>
<td></td>
</tr>
<tr>
<td>For environments with multiple patient identifiers, where accession # is not unique.</td>
<td></td>
</tr>
<tr>
<td>accession number+Pat id + DomainId</td>
<td>Supported with patient timeline in XDS environments.</td>
</tr>
<tr>
<td>san=&lt;acc&gt;&amp;pid=&lt;patientID&gt;&amp;DomainId=&lt; domainID&gt;</td>
<td></td>
</tr>
<tr>
<td>For environments with multiple patient identifiers, where accession # is not unique.</td>
<td></td>
</tr>
<tr>
<td>SUID</td>
<td>Not supported with patient timeline in XDS environments.</td>
</tr>
<tr>
<td>sui=&lt;study_instance_uid&gt;</td>
<td></td>
</tr>
<tr>
<td><strong>Query parameters on Centricity Enterprise Archive</strong></td>
<td><strong>Notes</strong></td>
</tr>
<tr>
<td>------------------------------------------------------</td>
<td>-----------</td>
</tr>
<tr>
<td>Pat id + authority + sui</td>
<td>Supported with patient timeline in XDS environments.</td>
</tr>
<tr>
<td>PatientId=&lt;Pat id&gt;&amp;authority&lt;Domain&gt;&amp;sui=&lt;sui id&gt;</td>
<td></td>
</tr>
<tr>
<td>Pat id +</td>
<td>Supported with patient timeline in XDS environments.</td>
</tr>
<tr>
<td>Domain Id +</td>
<td></td>
</tr>
<tr>
<td>Sui</td>
<td></td>
</tr>
<tr>
<td>PatientId=&lt;Pat id&gt;&amp;DomainId=&lt;Domain id&gt;&amp;sui=&lt;sui id&gt;</td>
<td></td>
</tr>
<tr>
<td>If specifying the global DomainId, make sure PixEnabled is true in DbConnections.config so that the global DomainId can be mapped to the correct EA authority.</td>
<td></td>
</tr>
<tr>
<td>Pat id + Domain Id + Sui</td>
<td></td>
</tr>
<tr>
<td>PatientId=&lt;Pat id&gt;&amp;DomainId=&lt;Domain id&gt;&amp;sui=&lt;sui id&gt;</td>
<td></td>
</tr>
<tr>
<td>Optional parameters that apply to either of the launch options</td>
<td>Not supported with patient timeline in XDS environments.</td>
</tr>
<tr>
<td>AE_title</td>
<td></td>
</tr>
<tr>
<td>AE_TITLE=&lt;AE_TITLE&gt;</td>
<td></td>
</tr>
<tr>
<td>Limits the patient level search and study level search to specified archive only.</td>
<td></td>
</tr>
<tr>
<td>We can pass multiple ae_title using back slash as a separator (including CPACS + EA archives).</td>
<td></td>
</tr>
<tr>
<td>The value for ae_title comes from the dbConnections configuration file of ZFP</td>
<td></td>
</tr>
<tr>
<td>Patient Name</td>
<td>Not supported with patient timeline in XDS environments.</td>
</tr>
<tr>
<td>pnm=&lt;PatientName&gt;</td>
<td></td>
</tr>
<tr>
<td>Format: Lastname, Firstname</td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Anonymous, GE33 will be returned when using:</td>
<td></td>
</tr>
<tr>
<td><a href="http://1.2.3.4/ZFP?mode=proxy#showlist&amp;un=ZFPUser&amp;pw=">http://1.2.3.4/ZFP?mode=proxy#showlist&amp;un=ZFPUser&amp;pw=</a>&lt;Password&gt;&amp;pnm=Anon*&amp;smo=CT\MR</td>
<td></td>
</tr>
<tr>
<td><a href="http://1.2.3.4/ZFP?mode=proxy#showlist&amp;un=ZFPUser&amp;pw=">http://1.2.3.4/ZFP?mode=proxy#showlist&amp;un=ZFPUser&amp;pw=</a>&lt;Password&gt;&amp;pnm=<em>GE</em>&amp;smo=CT\MR</td>
<td></td>
</tr>
<tr>
<td>Referring Physician</td>
<td>Not supported with patient timeline in XDS environments.</td>
</tr>
<tr>
<td>srp=&lt;ReferringPhysicianName&gt;</td>
<td></td>
</tr>
<tr>
<td>Format: Lastname, Firstname</td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Doctor, GE921 will be returned when using:</td>
<td></td>
</tr>
<tr>
<td><a href="http://1.2.3.4/ZFP?mode=proxy#showlist&amp;un=ZFPUser&amp;pw=">http://1.2.3.4/ZFP?mode=proxy#showlist&amp;un=ZFPUser&amp;pw=</a>&lt;Password&gt;&amp;srp=Doct*</td>
<td></td>
</tr>
<tr>
<td><a href="http://1.2.3.4/ZFP?mode=proxy#showlist&amp;un=ZFPUser&amp;pw=">http://1.2.3.4/ZFP?mode=proxy#showlist&amp;un=ZFPUser&amp;pw=</a>&lt;Password&gt;&amp;srp=<em>GE</em></td>
<td></td>
</tr>
</tbody>
</table>
## Query parameters on Centricity Enterprise Archive

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Query Example</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Modality</td>
<td><code>smo=&lt;Modality&gt;</code></td>
<td>Not supported with patient timeline in XDS environments.</td>
</tr>
<tr>
<td>Patient Sex</td>
<td><code>psx=&lt;PatientSex&gt;</code></td>
<td>Not supported with patient timeline in XDS environments.</td>
</tr>
<tr>
<td>Birth Date</td>
<td><code>pbd=&lt;BirthDate&gt;</code></td>
<td>Not supported with patient timeline in XDS environments.</td>
</tr>
<tr>
<td>Study Date</td>
<td><code>sdt=&lt;StudyDate&gt;</code></td>
<td>Not supported with patient timeline in XDS environments.</td>
</tr>
<tr>
<td>Status ID</td>
<td><code>sst=&lt;StatusID&gt;</code></td>
<td>Not supported with patient timeline in XDS environments.</td>
</tr>
</tbody>
</table>

### Notes:
- **Modality**
  - This parameter is used as a qualifier for the exam to limit search for Modality.
  - Example:
    - `http://1.2.3.4/ZFP?mode=proxy#showlist&un=USERNAME&pw=PASSWORD&pn=*GE*&smo=CT\MR`

- **Patient Sex**
  - This parameter is used as a qualifier for the exam to limit the search for patient sex. The value should be M for male, F for female, and O for other (according to the DICOM standard).

- **Birth Date**
  - This parameter is used as a qualifier for the exam to limit the search on patient Birthdate.
  - For example, Patient studies with birthdate of 10 July 1946 are returned with URL’s:
    - `http://1.2.3.4/ZFP?mode=proxy#showlist&un=USERNAME&pw=PASSWORD&pbd=10/07/1946`
    - `http://1.2.3.4/ZFP?mode=proxy#showlist&un=USERNAME&pw=PASSWORD&pbd=10-07-1946`
    - `http://1.2.3.4/ZFP?mode=proxy#showlist&un=USERNAME&pw=PASSWORD&pbd=10-Jul-1946`
    - `http://1.2.3.4/ZFP?mode=proxy#showlist&un=USERNAME&pw=PASSWORD&pbd=10/Jul/1946`

- **Study Date**
  - Not supported with patient timeline in XDS environments.

- **Study Description**
  - `sds=<StudyDescription>`
  - Study Description=Unspecified XA
  - Space is escaped with %20
  - Example:
### Query parameters on Centricity Enterprise Archive

<table>
<thead>
<tr>
<th>Document level launch</th>
<th>Patient Id + Domain Id + DocumentUid</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>PatientId=&lt;pat id&gt;&amp;DomainId=&lt;domain id&gt;&amp;DocumentUid=&lt;doc uid&gt;</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Auto View Latest document</th>
<th>AVL</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>aVL=ALL</td>
</tr>
</tbody>
</table>

When this parameter is specified, the latest document will be opened automatically when launching the Patient Timeline for a patient.

Auto View Latest can be used with standalone launch or with launch by PatientId, but it is not compatible with a document-level launch.

### Notes

Not supported with patient timeline in XDS environments.

Supported with patient timeline in XDS environments.

#### Query parameters on Centricity PACS*

<table>
<thead>
<tr>
<th>Patient level launch</th>
<th>Pat Id only</th>
</tr>
</thead>
<tbody>
<tr>
<td>ris_pat_id=&lt;patientID&gt;</td>
<td></td>
</tr>
</tbody>
</table>

Patient identifier + authority

ris_pat_id=<patientID>&authority=<domainID>

<table>
<thead>
<tr>
<th>Study level launch</th>
<th>accession number only</th>
</tr>
</thead>
<tbody>
<tr>
<td>ris_exam_id=&lt;acc&gt;</td>
<td></td>
</tr>
</tbody>
</table>

ZFP supports wild card so a site can set up.

ris_exam_id=F%, this will limit the user access to exam accession numbers that start with F.

accession number+Pat id

ris_exam_id=<acc>&ris_pat_id=<patientID>

Used where accession # is not unique.

accession number+Pat id + authority

ris_exam_id=<acc>&ris_pat_id=<patientID>&authority=<domainID>

For environments with multiple patient identifiers, where accession # is not unique.
<table>
<thead>
<tr>
<th>Query parameters on Centricity PACS*</th>
<th>URL query parameter</th>
</tr>
</thead>
<tbody>
<tr>
<td>SUID</td>
<td>study_instance_uid=</td>
</tr>
<tr>
<td>Optional</td>
<td>pat_name=</td>
</tr>
<tr>
<td>Patient Name</td>
<td>order_nbr=</td>
</tr>
<tr>
<td>Order Number</td>
<td>modality_code=</td>
</tr>
<tr>
<td>Modality</td>
<td>last_four_rpat_id=</td>
</tr>
<tr>
<td>Last four digits of Patient ID</td>
<td>study_dttm=</td>
</tr>
<tr>
<td>Study Date</td>
<td>procedure_desc=</td>
</tr>
<tr>
<td>Study Status</td>
<td>sts_stat=</td>
</tr>
<tr>
<td>Study Status Description</td>
<td>exam_stat_desc=</td>
</tr>
<tr>
<td>Display Search Controls</td>
<td>searchControl=</td>
</tr>
</tbody>
</table>

* These query parameters are not supported with Patient Timeline in an XDS environment.

3) Authentication and audit parameters supported on ZFP below:

<table>
<thead>
<tr>
<th></th>
<th></th>
<th>URL query parameter</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Anonymous user authentication using encrypted password.</td>
<td>pw</td>
</tr>
<tr>
<td>4</td>
<td>Username – anonymous user account used for authentication.</td>
<td>un</td>
</tr>
<tr>
<td>5</td>
<td>Custom parameter to audit username when anonymous or Windows authentication is used.</td>
<td>custom</td>
</tr>
</tbody>
</table>
4) Parameter validation for presence of special characters:

The following parameter values should not start with non-alphanumeric characters:

- Patient Id
- Accession Number
- Study Instance Uid
- Exam Id
- Patient Name
- Study Id
- LastFourPatId

3.2.5 ZFP URL Launch for MultiSite Command

1. For sites using the MultiSite command, the URL is going to look like:
   a. https://<zfpserver>/ZFP?mode=proxy#multiSite&siteId=<primarySiteId>&MRN=<primaryMrn>&AccNo=<primaryAccNo>|siteId=<comparisonSiteId1>&MRN=<comparisonMrn1>|siteId=<comparisonSiteId2>&MRN=<comparisonMrn2>
   i. Here <zfpserver> is specified at the customer site.
   ii. “|” is used as a separator for specifying the primary (one) and comparison information (multiple)

2. The URL, once it is formed, must be encrypted before launching it using ZFP as a multisite command is only supported with encrypted payload.
   a. If you use a multisite command with an unencrypted payload, the application will be re-directed to an error page.

The following parameters can be given while specifying the primary and comparison studies in the URL:

<table>
<thead>
<tr>
<th>Study</th>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>


### Study Parameter Description

<table>
<thead>
<tr>
<th>Study</th>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Primary</td>
<td>SiteId</td>
<td>Identifies a site (EaArchive). The following can be the input (provided they match the values given in the dbConnections.config file):</td>
</tr>
<tr>
<td></td>
<td>(mandatory)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- ArchiveDisplayName</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Archive</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- AeTitle</td>
</tr>
<tr>
<td>Primary</td>
<td>MRN</td>
<td>Specifies the patientId for the site.</td>
</tr>
<tr>
<td></td>
<td>(mandatory)</td>
<td></td>
</tr>
<tr>
<td>Primary</td>
<td>AccNo</td>
<td>Specifies the accession number for the patient.</td>
</tr>
<tr>
<td></td>
<td>(mandatory)</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Study</th>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Comparison</td>
<td>SiteId</td>
<td>Identifies a site (EaArchive). The following can be the input (provided they match the values given in the dbConnections.config file):</td>
</tr>
<tr>
<td></td>
<td>(mandatory)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- ArchiveDisplayName</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Archive</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- AeTitle</td>
</tr>
<tr>
<td>Comparison</td>
<td>MRN</td>
<td>Specifies the patientId for the site.</td>
</tr>
<tr>
<td></td>
<td>(mandatory)</td>
<td></td>
</tr>
</tbody>
</table>

Below are a few basic scenarios that will cause the URL to become invalid and ZFP to throw a pop-up error:

- Failing to provide the mandatory parameters (primary or comparison)
- Providing siteId which is not mentioned in dbConnections
- Providing extra parameters in the URL (while specifying either primary or comparison)
- Providing more than one primary info while forming the URL

There will be more validation in place, which will result in a generic error being thrown on the ZFP UI. To get details around the error, look at the browser console; details will be logged there.

### 3.2.6 ZFP Launch Configurations (Information for Calling Application)

1. For sites that are using anonymous authentication (that is, they are not using the SAML or Windows Authentication option), they need to agree on a password. The value is an encrypted string.
Parameter | Value | Default value on ZFP
---|---|---
Password (pw parameter passed in the URL) | This value is sent through the URL to launch ZFP. The third-party application and ZFP use the 'Private Key' value to encrypt and decrypt the password. | ""

2. ZFP lets you securely integrate from the calling application by providing an option for dynamic encryption. If the integration uses encrypted payload, then the additional configurations must be synchronized between the two applications. This ensures that the calling application can encrypt and ZFP can decrypt the same. Refer to Appendix D for a sample URL created with the below algorithms supported.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
<th>Default value on ZFP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Encryption Algorithm</td>
<td>AES or 3DES (only for backward compatibility)</td>
<td>AES</td>
</tr>
<tr>
<td>Padding Mode</td>
<td>PKCS7, None, Zeros, ANSIX923 or ISO10126</td>
<td>PKCS7</td>
</tr>
<tr>
<td>Cipher Mode</td>
<td>CBC, ECB, OFB, CFB or CTS</td>
<td>CBC</td>
</tr>
<tr>
<td>Time Format</td>
<td>UTC</td>
<td>UTC</td>
</tr>
<tr>
<td>URL Encoding</td>
<td>Ensure that URL encoding is done before launching ZFP from a third-party application.</td>
<td>Required</td>
</tr>
</tbody>
</table>
| Private Key | This key is used for encryption and decryption of URL parameters. This key must be 16 or 24 characters in length. It needs to be the same in ZFP and the third-party application. | ""

Fig : Integration Worksheet

When configured with a Centricity PACS back end, when the user launches ZFP from an external application, if the user password expires then the user is required to update his password using the Centricity Administration Tool (CAT) of Centricity PACS. The user will not be redirected to the Change Password screen in this scenario.

### 3.3 ZFP Java Script Launch

ZFP can be launched programmatically from external applications using functions provided in the JavaScript file WebViewerProxy.js. For Microsoft Internet Explorer (IE) v8, you can use WebViewerProxy_IE8.js.

These files are in the ZFP install directory: ZFP/nocache

Sample HTML reference code to invoke the JavaScript file: `<script type="text/javascript" src=http://ZFP_URL/ZFP/nocache/WebViewerProxy.js></script>`

Javascript launch supports the following function interfaces:
<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
<th>Sample Code</th>
</tr>
</thead>
</table>
| Initialize(window, strUrlLocation, strUsername, strPassword, strCustom) | The Initialize command prepares the WebViewerProxy object to start a ZFP Viewer session. The ‘window’ argument should contain the HTML window that must host the ZFP Viewer. The ‘window’ argument is either an IFrame object or a window object. The ‘username’ and the ‘password’ argument can be used or left empty. In the latter case, the ZFP Viewer uses its default login procedure. The ‘custom’ argument can log information to the usage log. | var viewerWindow = window.open("about:blank", "ZfpViewer", "width=1500,height=800,location=no,menubar=no,titlebar=no,left=150,top=150");
var webviewerproxy = new WebViewerProxy();
| Execute(strCommand) | strCommand supports 3 commands:
• view
• viewall
• showlist | Execute(view&pnm=Watson)
Execute(viewall&pnm=Watson)
Execute(showlist&pnm=Watson) |
| OnConnectionChange() | When the connection status of the ZFP Viewer changes, the third party can be notified by setting the OnConnectionChange method to a function within the third-party code. When the function is called, the third party can then call the IsConnected function to check the status. This function should be set before calling Initialize(). | function OnConnectionChange(){
  if (webViewerProxy.IsConnected()) {
    alert("Connected!");
  } else {
    alert("Disconnected!");
  }
}
webViewerProxy.onconnectionChange = OnConnectionChange;
webviewerproxy.Initialize(WVWindow, "localhost/ami", "MyUsername", "Mypassword", "log this"); |
Chapter 3: ZFP Inbound Launch

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
<th>Sample Code</th>
</tr>
</thead>
</table>
| IsConnected()     | This function returns a Boolean to the third-party application. If the value is True, then the ZFP Viewer is ready to receive commands through the Execute interface. If the value is False, then the ZFP Viewer will not respond to commands sent through the Execute command. If the value is False and the Initialize function is called, the third party can continue to wait for a connection or recreate the webViewerProxy object through a new WebViewerProxy() and call Initialize again. | function OnConnectionChange(){
  if (webViewerProxy.IsConnected()) {
    alert("Connected!");
  } else {
    alert("Disconnected!");
  }
} |

Notes

1. **View Command:** If the query result contains more than one record, then ZFP Viewer will stay in the Search results instead of going to the viewer. If the query result contains exactly one record, then ZFP Viewer displays the record directly in the viewer.

2. **ViewAll Command:**
   - The viewall command instructs ZFP Viewer to show the studies, selected with the query, directly in the viewer. Compared to the view command, ZFP Viewer will always switch to the viewer with viewall command.
   - User can control the display of multiple patient selection messages by passing in the following parameter: DisplayMultiplePatientMessage

**Valid Values:** True or False (Default is True)

The message will not be displayed to the user if the parameter value is set to ‘False’ (see example below) and the most recent study will get displayed with the title bar displaying patient demographic information. The remaining results are displayed in the Series Selector and Navigator.

**Example (Centricity EA)**
viewall&pnm=Watson*&DisplayMultiplePatientMessage=false

**Example (Centricity PACS)**
viewall&pat_name=Watson%&DisplayMultiplePatientMessage=false

- ‘primarySui’ Parameter use with ViewAll Command:
  The primarySui parameter is meant to be used with the viewAll command along with other query parameters. This parameter works for both EA and CPACS back ends.
  When the primarySui parameter is provided, the value should be a single study instance UID. The study matching this UID will be marked as the primary. All other studies matching the remaining query parameters will be loaded as comparison studies.
  For example, the EA backend command using primarySui:
viewall&primarySui=1.2.34.567.891011&sui=1.2.98.456.33\1.2.4562.2332 will load the study with UID 1.2.34.567.891011 as the primary and the studies with UID 1.2.98.456.33 and UID 1.2.4562.2332 as comparisons. In this case, we have specified multiple sui values so we can load multiple studies specifically by their unique IDs. Note that the IDs are separated by the \ (forward slash).
Chapter 3: ZFP Inbound Launch

For example, the CPACS command using primarySui:
viewall&primarySui=1.2.34.567.891011&ris_pat_id=1234 will load the study with UID 1.2.34.567.891011 as the primary and all other studies for patient ris_pat_id 1234 as the comparisons. In this case, it is possible that the patient with ris_pat_id equal to 1234 has multiple studies. Those studies will be available as comparisons along with the primary study specified by primarySui. With CPACS, ZFP does not support specifying multiple study instance UIDs.

3. Showlist command: The showlist command instructs the ZFP Viewer to perform a query and to display the query result in the study results.

Note: Javascript launch is not supported with Patient Timeline in an XDS environment.

3.4 Encryption of URL Parameters for ZFP Launch from External Applications

1. Open an authentication tool by running the following EXE file from the folder Program Files (x86)\GE HealthCare\ZFP\AuthenticationTool:

GEHealthcare.ZFP.AuthenticationConsole.exe

2. When prompted to enter a choice, enter 1 for encrypting a URL token.
3. Enter the plain text password to be encrypted and press Enter.
4. Copy the encrypted password from this window into the URL for the ‘pw’ parameter.

3.5 Decryption for Payload

1. Open an authentication tool by running the following EXE file from the folder Program Files (x86)\GE HealthCare\ZFP\AuthenticationTool:

GEHealthcare.ZFP.AuthenticationConsole.exe

2. When prompted to enter a choice, enter 2 for decrypting a URL token.
3. You are prompted to enter a token to decrypt. Enter the encrypted URL parameters and press Enter. You will see a decrypted value of the URL parameters.

3.6 Generating Hexadecimal Key for Open SSL Encryption

1. Open an authentication tool by running the following EXE file from the folder  Program Files (x86)\GE HealthCare\ZFP.AuthenticationTool:

GEHealthcare.ZFP.AuthenticationConsole.exe

2. When prompted to enter a choice, enter 3 for Hex Private Key.
3. When prompted to enter a key to generate a Hex key, enter the plain text key and press Enter. A hex value of the plain text key displays.

3.7 ZFP Launch Configurations (Information for GE Service Engineer)

Centricity Web Replacement

When replacing Centricity Web with ZFP, the URL gets updated as follows:

Centricity Web URL


would become as follows in ZFP:
Chapter 3: ZFP Inbound Launch

https://<ZFPServerIP>/ZFP?mode=proxy#view&accessionNumber=DH1085993&patId=349574&un=TestUser&pw=GTYWP:UR&^GFDLKSD

**ZFP Server Configurations**

Follow the steps in the *Centricity Universal Viewer Zero Footprint Client Installation and Upgrade Manual* to configure URL authentication.

### 3.8 ZFP Time-out in URL Launch Mode

By default, the ZFP viewer locks out a user session according to the IIS time-out configuration when launched using URL Launch. Contact your GE Service Representative for more information about setting the time-out value.

The ZFP viewer also includes a configuration in the AppSettings.config file that allows ZFP in URL and Java Script based launch mode to avoid timing out and locking out the user session. The configuration is as follows:

```xml
<TimeoutUsingProxy>false</TimeoutUsingProxy>
```

- If the value is true, then the ZFP Viewer times out according to the IIS configuration.
- If the value is false, then the ZFP Viewer sets a time-out for all URL and Java Script based launch mode sessions that is the maximum allowed by IIS. The maximum allowed time-out is one year.

**Note:** This is not supported with Patient Timeline in an XDS environment.

### 3.9 ZFP Outbound Launch

Refer to the *Centricity Universal Viewer Zero Footprint Client Administration Manual* for details on how ZFP can be configured to launch third-party applications.
Appendix A: Authentication

Anonymous user authentication using static token (encrypted password)
ZFP can be configured with an external application to launch in static token mode with encrypted password. In this scenario, the URL should be appended with two query string parameters, as shown below.

Example URL
https://<ZFPServerIP>/ZFP?mode=proxy#view&<study_search_parameters>&un=<user_name>&pw=<authentication_key>&custom=CustomUser

where
study_search_parameters = query parameters to open a study
un = User name for auditing purpose
pw = Encrypted authentication key
custom = Custom string that will be used for audit logging

Anonymous user authentication using dynamic token
ZFP can be configured with an external application to launch using a dynamic token. In this scenario, the URL should be appended with the following parameters:

Example URL
https://<ZFPServerIP>/ZFP?mode=proxy#pl=<encrypted_payload>

Example URL (XDS-based Patient Timeline)
https://<ZFPServerIP>/ZFP/PatientTimeline?pl=<encrypted_payload>

where
encrypted_payload = encrypted query parameters and/or username and password.

Notes
• The ‘un’ parameter is any generic user name. This is required for logging. The ‘pw’ parameter is any password chosen by the admin and AES-encrypted in the Security.config file on the ZFP server for the below key. This password needs to be shared with the third party application so it can be configured for creating the ZFP URL.

<UrlAuthentication enabled="true" token="<aes-encrypted-token>" />
Appendix B: Open Desktop Integration with Microsoft IE 8 and 9

IE8 and IE9 Support
Customers using ZFP with IE8 or IE9 must have Legacy Browser Component installed. The Legacy Browser Component enables optimal performance when running on IE8 or IE9.

ZFP with Legacy Browser Component
For Internet Explorer 8 and 9 with Legacy Browser Component installed, ZFP loads inside of an ActiveX control that leverages the CefSharp3 library which loads ZFP using Chromium. This allows ZFP to achieve functionality and performance on par with Google Chrome browser while still using Internet Explorer.

Launching ZFP with Self-Signed Certificate in an “iFrame” in IE8 or IE9 with LBC
To display DICOM images within an application, the application can embed ZFP in an iFrame. For external applications, embedding ZFP with a self-signed certificate in an iFrame in IE versions 8 or 9 with Legacy Browser Component (LBC) installed must follow these steps:

1. Install the self-signed certificate under Trusted Root Certification Authorities.
   a. In Internet Explorer, browse to the ZFP site with self-signed certificate. Internet Explorer will display a Certificate Error at the right-hand side of the address bar. Click Certificate Error and select View certificates.
   b. Click Install Certificate, and then in the wizard, click Next.
   c. Select Place all certificates in the following store.
   d. Click Browse, select Trusted Root Certification Authorities, and click OK.
   e. From the wizard, click Next and then Finish.

2. Add the ZFP URL to a Trusted Sites list.
Appendix B: Open Desktop Integration with Microsoft IE 8 and 9

a. Open Internet Options from IE settings and select Security > Trusted sites > Sites.
b. Enter the ZFP URL in the Add this Website to this zone text box.
c. Click Add and then Close.

3. Close the IE instance and re-launch IE. The self-signed certificate should now be trusted. The external application should be able to launch ZFP in the iFrame and display images.
Introduction and Scope

ZfpApiTool is a 32-bit executable to launch ZFP without the need to know the syntax of the ZFP web URL mode and without programming experience required by the Java Script-based launch mode. Certain applications, such as Centricity RIS-I, use this tool to integrate with ZFP. ZfpApiTool supports Inbound ZFP launch with Microsoft Internet Explorer browser versions IE8 and IE9 (with Legacy Browser Component), IE10 and IE11.

This appendix describes the installation and configuration of ZfpApiTool and its different operational modes.

Test Mode

For testing ZFP Java Script-based launch mode and ZfpApiTool itself, a dialog-based GUI allows you to execute various commands to ZFP by passing any combination of supported query parameters. User credentials and server name have to be specified in advance.

Batch Mode

Using command line parameters, ZfpApiTool is able to start ZFP in Java Script based launch mode by passing any possible combination of query parameters. A mechanism prevents multiple instances of ZfpApiTool, and thus ZFP, from being started as long as server, username, and encrypted password are unchanged. Instead, all calls are processed by the sole instance.

Note:
This document does not include information about how to create users in Centricity Enterprise Archive/Centricity PACS as well as the installation and configuration of those two products. Contact your GE Service Representative for more information.

Test Mode

For testing ZFP Java Script-based launch mode and ZfpApiTool, a dialog-based GUI allows you to execute commands to ZFP passing any combination of query parameters as supported by the selected back end.

To start ZfpApiTool in test mode, launch it without specifying parameters. In the command line, navigate to the directory in which the ZfpApiTool is saved and enter the following:

ZfpApiTool.exe

It starts as a window covering the primary screen, and it displays the Query dialog box, as depicted in Figure 1. The user can select whether the back end underneath ZFP is Centricity PACS or EA. This modifies the query parameters displayed in the lower part of the dialog box since different back ends support different parameters, including their names.

The server address (host name or IP of the machines where ZFP is running) and the user name/encrypted password of a valid ZFP user account are mandatory with the command to be issued.

NOTE: The password should be encrypted. Refer to section 3.4 Encryption of URL Parameters for ZFP Launch from External Applications.
Appendix C: ZFPApiTool

ZfpApiTool also supports all 3 commands explained 26:

Viewall
View
Showlist

Note:
If the number of studies found by a specific query exceeds a pre-configured limit, an error message is displayed and the operation is aborted. Ask your administrator or other qualified personnel about the maximum number of studies a query may result in. This number can be increased, if needed.

The query parameters in the lower part of the dialog are not discussed here. Their names are self-explanatory and depend on the selected back end.

Figure: Test mode query dialog —Centricity PACS vs. Centricity Enterprise Archive back end

Same Query parameters are used with Centricity EA and Centricity PACS back ends as with URL and Java Script launch. Once the appropriate information is entered into the query dialog, press Apply to execute the query. To perform further commands, press the Query button in the left lower corner of the ZfpApiTool window to bring the
Appendix C: ZFPApiTool

query dialog up again. Furthermore, the status bar to the right side of the Query button contains the syntax of the latest command that has been issued to ZFP. For an example see Figure 4.

Figure: Test mode status bar

Batch Mode

Using command line parameters, ZfpApiTool can start ZFP in Java Script-based launch mode by passing any possible combination of query parameters. A mechanism prevents multiple instances of ZfpApiTool, and thus ZFP, from starting as long as the server, user name and encrypted password are unchanged. Instead, all calls are forwarded and then processed by the single instance. This instance keeps a ZFP Proxy object, and does not consume any time to load ZFP and login after the first operation completes successfully.

The following illustrates the command line syntax of ZfpApiTool in batch mode:

```
ZfpApiTool.exe -batch -user <username> -passwd <encrypted password> -https <true/false> -server<ZFP server hostname or IP> -mode <command> [-<queryparam1name> < queryparam1queryvalue> -<queryparam2name> < queryparam2queryvalue> ...]
```

Where:

- The parameter -batch indicates that ZfpApiTool shall be started in batch mode and the following command line shall be parsed.
- The -user parameter needs to be followed by the user name of a valid ZFP user, -passwd by the encrypted password (Refer Section 3.4 Encryption of URL Parameters for ZFP Launch from External Applications) of this particular account as well as -server by the hostname or IP address of the machine where the ZFP is running.
- The –https parameter is used to select whether or not the communication will use HTTPS communication. By default, if –https parameter is not provided in the command line, the ZFP API Tool will use HTTPS.
- The <command> value after -mode can be one of the following:
  - showlist
  - view
  - viewall

See the command description in above section for details.

Example (Centricity EA)

```
ZfpApiTool.exe –batch –user web –passwd Bl3thb3b0cTZGPUGR2ZQtTcMXdGaSrdT –server testsrv1 –mode viewall –pnm Watson* -DisplayMultiplePatientMessage false
```

Example (Centricity PACS)

```
ZfpApiTool.exe –batch –user web –passwd Bl3thb3b0cTZGPUGR2ZQtTcMXdGaSrdT –server testsrv1 –mode viewall –pnm Watson% -DisplayMultiplePatientMessage false
```
Appendix C: ZFPApiTool

**Examples**

After the command value, zero to n query parameter name/value pairs can be specified. This depends on the chosen command as well as your needs to query the ZFP worklist before executing the appropriate command.

Following are some examples:

ZfpApiTool.exe –batch –user web –passwd Bl3thb3b0cTZGPUGR2ZQttCcMXdGsrD –server testsrv1 –mode showlist –ris_pat_id 123456 –modality_code CT

The above command can be used to display the query result in the ZFP Data Selector containing all the CT studies belongs to the patient with RIS patient ID 123456, or:

ZfpApiTool.exe –batch –user web –passwd Bl3thb3b0cTZGPUGR2ZQttCcMXdGsrD –server testsrv1 –mode viewall –ris_pat_id 123456 –modality_code CT

The above command can be used to open all CT exams belonging to the patient with the RIS patient ID 123456 for viewing.

When using the view Command, the query specified by the query parameter pair needs to return exactly one result so that this study can be opened for viewing. If more than one result exists, the study result is opened and it displays all the returned studies.

**Browser Support**

ZfpApiTool supports IE8 and IE9. Clients using ZFP with IE8 or IE9 must have either Legacy Browser Component installed.

**Installation**

To use the ZfpApiTool, follow the steps below:

1. Copy the following files from the ZFP server to the server of your choice;
   a.  `<ZFP Media Root>\Universal Viewer Zero Footprint\Prerequisites\vcredist_x86\2012\vcredist_x86.exe`
   b.  `C:\Program Files (x86)\GE HealthCare\ZFP\bin\ZfpApiTool.exe`

2. Install the VS2012 Redistributable C++ libraries on the target machine by executing the following file:
   `<target directory>\vcredist_x86.exe`

Internet Explorer version 8 and 9 users: Customers using ZFP with IE8 or IE9 must have Legacy Browser Component installed.

(Refer Appendix B  Open Desktop Integration with Microsoft IE 8 and 9

You are ready to use the ZfpApiTool. You can run the tool from the command line in test mode or batch mode.
Appendix D: Creating a ZFP URL

To create a ZFP URL with encrypted payload:

The encryption algorithm used is AES, Cipher mode is CBC and Padding mode is PKCS7.

The following test query string is used for encryption:
view&sui=1.2.3.45.6789.0&pw=yP9a5foBZtU=&un=testUser

1. Append a pipe "|" and the current UTC date time in the format "yyyyMMddHHmmss". This will result in:
   view&sui=1.2.3.45.6789.0&pw=yP9a5foBZtU=&un=testUser|20131213134546

2. Prepend a random data block (128 bits = 16 bytes) at the beginning of the message. This is important for AES encryption, without this you will not be able to generate the encrypted string.
   You can use the following command:
   
   `openssl rand -out randomblock.txt 16`

3. To encrypt, the input length should always be in multiples of 16 bytes. If the total input length after adding random block is not a multiple of 16, then add the missing bytes. Next, pad the string to the next full block size (16 bytes) for CBC. In PKCS7 padding, this means that the value used for padding will be the amount of missing bytes. In this case there are 13 bytes left until a full block, so pad with the hex value "0d". To get the Hex representation of the string, use any tool or plug-in.

Here is a hex representation of the string after this:

```
00000000 83 95 62 13 b7 cd a5 fa 10 3f 02 0e ce 67 07 c8 |..b......?...g..|
00000010 76 69 65 77 26 73 75 69 3d 31 2e 32 2e 33 2e 34 |view&sui=1.2.3.4|
00000020 35 2e 36 37 38 39 2e 30 26 70 77 3d 79 50 39 61 |5.6789.0&pw=yP9a|
00000030 35 66 6f 42 5a 74 55 3d 26 75 6e 3d 74 65 73 74 |5foBZtU=&un=test|
00000040 55 73 65 72 7c 32 30 31 33 31 32 31 33 34 |User|2013123134|
00000050 35 34 36 0d 0d 0d 0d 0d 0d 0d 0d 0d 0d 0d 0d 0d |546.............|
```

4. Save the padded string in a file called to_encrypt.txt.

5. Use the ZFP Authentication tool to generate the encryption key in Hexadecimal format. Plain text key value is not accepted in open ssl encryption. Refer to section 3.6 to generate a hex key. This will be used in the open ssl command to generate the encrypted value.

6. Use the following command with openssl to encrypt the string using AES in CBC mode:
   
   `openssl enc -aes-256-cbc -in to_encrypt.txt -out result.enc \
   -K 69A721F2FE50909FA25571B3227DCAB731491AA55BEA2776F293F4A6E0C898B8 \
   -p -nosalt -iv 0 -nopad`
   
   This will result in result.enc containing the encrypted string.

7. Make sure the encrypted string is base64 encoded:
Appendix D: Creating a ZFP URL

```
cat result.enc | base64 > result-base64.txt
```

The base64 encoded string will be:

```
ZhHQZeDGdawywnNmID320Tpcb4LlNhmkLRB+i7DJjvSq5D7L9VWpPZeJfDLb9cbokjVyUvW5khX6QTh1bX2VA4/9w0pewdjMp08PekDJD3LQSdvwM2st/EjWe7ei5iNI
```

8. Generate the URL encoded value. Once the URL encoding is done, use the Authentication tool to generate the decrypted value, and then verify if the encrypted value is the same as the decrypted value. Refer to section 3.5 to decrypt.

```
ZhHQZeDGdawywnNmID320Tpcb4LlNhmkLRB%2Bi7DJjvSq5D7L9VWpPZeJfDLb9cbokjVyUvW5khX6QTh1bX2VA4%2F9w0pewdjMp08PekDJD3LQSdvwM2st%2FEjWe7ei5iNI
```

The encryption key (-K parameter for openssl) is generated from a password. The code uses AesCryptoServiceProvider for encryption and decryption.

```
```

The method for creating the encryption key is Rfc2898DeriveBytes.GetBytes

```
```

The parameters for the method are GetBytes(32).

Hence, the resulting ZFP URL to launch from external application is as follows:

```
http://<ZFPServerIP>/ZFP?mode=proxy#pl=ZhHQZeDGdawywnNmID320Tpcb4LlNhmkLRB%2Bi7DJjvSq5D7L9VWpPZeJfDLb9cbokjVyUvW5khX6QTh1bX2VA4%2F9w0pewdjMp08PekDJD3LQSdvwM2st%2FEjWe7ei5iNI
```
Appendix E: Using Cryptography.dll to Encrypt URL

To generate the encrypted authentication key or encrypted payload (containing all the study parameters, authentication key and user name) programmatically, ZFP provides the library ZFP.Cryptography.dll. This library could be found at “Program Files (x86)\GE HealthCare\ZFP.AuthenticationTool\ZFP.Cryptography.dll” on the ZFP server.

ZFP.Cryptography.dll provides two interfaces and classes, as shown below.

Interfaces
ICryptographyConfiguration – This interface is to set up the configuration parameters for encryption/decryption. The parameters include secret key, cipher mode, and padding mode.

ICryptographyService – This interface provides methods for encryption and decryption.

Classes
CryptographyConfiguration – This class initializes the cryptography parameters secret key, cipher mode, and padding mode.

CryptographyService – This class has implementation methods for encryption and decryption. This class has two methods: Encrypt(string dataToEncrypt), Encrypt(string dataToEncrypt, bool appendTimestamp) and Decrypt(string dataToDecrypt).

Usage
The following is the code snippet explaining how to programmatically encrypt and decrypt a text (written in C# programming language):

```csharp
//Initialize the cryptography parameters. Below parameters must be same as what was configured in Security.config file on ZFP server for the keys "PrivateKey", "CipherMode" and "PaddingMode".
string _secretKey = "secretkey1234567", _cipherMode = "CBC", _paddingMode = "PKCS7";

//Create an instance of CryptographyConfiguration class.
ICryptographyConfiguration _cryptographyConfiguration = new CryptographyConfiguration(_secretKey, _cipherMode, _paddingMode);

//Create an instance of CryptographyService class.
ICryptographyService _cryptographyService = new CryptographyService(_cryptographyConfiguration);
```
Appendix E: Using Cryptography.dll to Encrypt URL

//Create string with study parameters, the user name and the password (pw). Obtain the password (pw) from site administrator.
var _urlToEncrypt = "sui=1.2.34343.12.3233&un=testUser&pw=TestZFP@123!";

//Call CryptographyService.Encrypt method to get encrypted payload. This will append UTC timestamp (yyyyMMddHHmmss) to the original string to be encrypted and then performs encryption.
var _encryptedPayload = _cryptographyService.Encrypt(_urlToEncrypt);

/* If the requirement is to encrypt the string without appending the timestamp then the above line of code will change to: */
var _encryptedPayload = _cryptographyService.Encrypt(_urlToEncrypt, false);

Note:
The encrypted string returned by the Encrypt() method mentioned in the above example always returns a URL-encoded string.
Appendix F: CWeb to ZFP Migration

IIS Configuration for URL Redirect

Some sites launch studies from their EMR system using a URL link to Centricity Web (CWeb). After replacing CWeb with the ZFP server, the links in the EMR system are invalid. For example, links to the old CWeb Server may be stored in the EMR database. The procedure below configures IIS to rewrite a CWeb URL as a ZFP URL. This ensures that previously created links will continue to work without modifying the existing links in the database. The redirection happens just in time when a user opens the existing CWeb link via the EMR or other third-party application.

1. Install the URL Rewrite module for IIS 7.0. It is available from Microsoft at http://www.iis.net/downloads/microsoft/url-rewrite
2. Once the URL Rewrite module is installed, open IIS.
3. Navigate to site running ZFP, for example, Default Web Site or ZFPDefault.
4. Right-click on the site running ZFP and select Explore.

5. Windows Explorer opens in the root of the site running ZFP. Open the Web.config file in Notepad.
6. Find the <system.webServer> section in the Web.config file. If it does not exist, add it as the last section in the <configuration> section. For example,
   
   ```xml
   <configuration>
     ...
     ...
     ...
     <system.webServer>
     </system.webServer>
   </configuration>
   ```

7. Add the following within the <system.webServer> section. These configurations are explained later.
   
   ```xml
   <system.webServer>
     <rewrite>
     <rules>
   ```
Appendix F: CWeb to ZFP Migration

<clear />

<!-- HTTPS Redirect : All requests are redirected to HTTPS -->
<rule name="ssl" stopProcessing="true">
    <match url="(.*)" />
    <conditions logicalGrouping="MatchAll" trackAllCaptures="false">
        <add input="{HTTPS}" pattern="^OFF$" />
    </conditions>
    <action type="Redirect" url="https://DicomViewer.ZFP/{R:1}" redirectType="SeeOther" />
</rule>

<!-- CWeb Redirect : Convert CWeb requests to ZFP requests -->
<rule name="EMRweb2zfp" enabled="true" stopProcessing="true">
    <match url="^ami/(\[html/webviewer.htm\]+)" />
    <conditions logicalGrouping="MatchAny" trackAllCaptures="false">
        <add input="{QUERY_STRING}" pattern="(\[view\]+)&(\[_0-9a-z=\]+)&(\[pw=CANal2\]+)&(\[_0-9a-z=\]+)" />
        <add input="{QUERY_STRING}" pattern="(\[viewall\]+)&(\[_0-9a-z=\]+)&(\[pw=CANal2\]+)&(\[_0-9a-z=\]+)" />
        <add input="{QUERY_STRING}" pattern="(\[showlist\]+)&(\[_0-9a-z=\]+)&(\[pw=CANal2\]+)&(\[_0-9a-z=\]+)" />
    </conditions>
    <action type="Redirect" url="ZFP?mode=proxy#{C:1}&{C:2}&pw=kgo\n\n\nappendQueryString="false" />
</rule>

8. If the site is not using HTTPS, remove the HTTPS Redirect <rule> section. It can be added back later.
9. Edit the Web.config file to replace all occurrences of “CANal2” with the CWeb password used at the site.
10. Edit the Web.config file to replace “kgoknOf\ntndOkmbmX6Z\nHp” with the encrypted ZFP password. See section 3.4 Encryption of URL Parameters for ZFP Launch from External Applications for more information on encrypting the password.
12. Restart IIS.

The ZFP public IP address must be the previously-used CWeb public IP address. The previously-used DNS records for the CWeb server must point to the DNS record of the ZFP server. In a load-balanced environment the DNS pointer should refer to the public Fully Qualified Domain Name (FQDN).

After restarting IIS, the two <rule> sections above accomplish the following.

HTTPS Redirect

The <rule> section above titled HTTPS Redirect converts all requests to ZFP to HTTPS. For example, if a request to the server is http://192.0.0.1/zfp, the rule redirects the URL to https://DicomViewer.ZFP/zfp. Everything after the http://192.0.0.1/ remains unchanged. This rule forces all requests to use a secure HTTP connection. Edit the file Web.config to replace “DicomViewer.ZFP” with either the IP address or the FQDN of the ZFP server. If the site is not using secure HTTP communication, remove the <rule> section.

CWeb Redirect

The main difference between CWeb URLs and ZFP URLs is the password (pw) parameter. The ZFP URL password parameter must be encrypted, unlike CWeb, which uses a plain text value for the password. See section 3.4 Encryption of URL Parameters for ZFP Launch from External Applications for more information on
Appendix F: CWeb to ZFP Migration

crypting the password. As described in the steps above, make sure to edit the Web.config file to replace “CANal2” with the CWeb password and to replace “kgokmnOftndOkmbmX6ZgqHP” with the encrypted ZFP password.

The <rule> section above titled CWeb Redirect converts the CWeb query string into a query string recognized by ZFP. By modifying the CWeb and ZFP passwords in the example above, the rules should work for most sites. If you have trouble, see http://www.iis.net/learn/extensions/url-rewrite-module/creating-rewrite-rules-for-the-url-rewrite-module. Below are details about the rules that may help diagnose the problem.

The rule is written to convert a valid CWeb link to a valid ZFP link. Specifically, the rule uses regular expressions to replace portions of the original URL. For example,

http://cweb.com/ZFP?mode=proxy#showlist&un=CWEBUSER&pw=kgokmnOftndOkmbmX6ZgqHP&pid=1234&san=Acc56 and

http://cweb.com/ami/html/webviewer.htm?view&someParam=value&un=CWEBUSER&pw=CANal2&sui=1.2.3.456 becomes
http://cweb.com/ZFP?mode=proxy#view&someParam=value&un=CWEBUSER&pw=kgokmnOftndOkmbmX6ZgqHP&sui=1.2.3.456

The rule redirects a URL, which includes ami/html/webviewer.htm and which the query string matches the pattern one of the following patterns:

\(([\text{view}\]+)&(\(_0-9a-z=\)+)&(\[pw=CANal2\]+)&(\(_0-9a-z=\)+)\)
\(([\text{viewall}\]+)&(\(_0-9a-z=\)+)&(\[pw=CANal2\]+)&(\(_0-9a-z=\)+)\)
\(([\text{showlist}\]+)&(\(_0-9a-z=\)+)&(\[pw=CANal2\]+)&(\(_0-9a-z=\)+)\)

Parts of the pattern are wrapped in parentheses and square brackets. The parentheses allow the URL Rewrite module to group parts of the CWeb URL and reuse them in the ZFP URL. The <action> section of the rule describes how the URL is rewritten using the pattern:

ZFP?mode=proxy#{C:1}&{C:2}&pw=kgokmnOftndOkmbmX6ZgqHP&{C:4}.

Notice the {C:1}, {C:2} and {C:4} parts. These correspond to first, second, and fourth sets of parentheses in the URL match pattern.

The first group identified as {C:1} matches the command, either view, viewall, or showlist. The {C:2} and {C:4} groups match multiple occurrences of key=value pairs using the pattern \(_0-9a-z=\)+). The regular expressions assumes that there are key=value pairs before and after the pw=CANal2 part of the URL. If this is not the case for customer site, the regular expression needs to be modified to fit the customer needs.
Appendix G: Proxy Launch from Non-Web-based Application

Introduction and Scope
ZFP URL launch described in section 3.2 allows launching ZFP for a study, but requires launching a new browser window, and initializing the ZFP viewer, which may take a few seconds.

ZFP JavaScript launch described in section 3.3 allows pre-initializing the ZFP viewer and opening studies in an already initialized viewer instance, resulting in shorter startup time.

Unfortunately, some EMR or third-party applications do no allow embedding ZFP, which leaves the URL launch option. As a result, a new ZFP instance is initialized every time the user launches a new study, causing unwanted delay.

To fix this problem, you can launch a study in an already initialized ZFP instance, without requiring the calling application to embed ZFP or use WebViewerProxy.

Outline
The solution presented in this section is a lightweight tool (a small executable or a Visual Basic script) that will load a new study in an existing ZFP instance whenever possible. If no ZFP instance is available, the script starts a new browser instance to initialize ZFP.

Limitations

- This tool works only on client machines running the Windows operating system.
- This tool works only with browsers IE8 and IE9 (with LBC), IE10, and IE11.
- A small executable or VB script file must be copied to every client machine.
- Users need permission to run the executable or run VB script (i.e. execute cscript.exe).

Setup Procedure
3. On the client machine, copy either “IEExternalViewerLaunch.exe” or “IEExternalViewerLaunch.vbs” from C:\Program Files (x86)\GE HealthCare\ZFP\nocache to a known location.
4. On the client machine, configure the EMR or third-party application to launch the executable or VB script with the ZFP URL as an argument:
   If using the executable:
   
   C:\Temp> IEExternalViewerLaunch.exe
   "https://<ZFPserverIP>/ZFP?mode=proxy#viewall&un=<user_name>&pw=<authentication_key>&sui=<study_search_parameters>"
Appendix G: Proxy Launch from Non-Web-based Application

If using the VB script:

C:\Temp> cscript.exe IEExternalViewerLaunch.vbs
  "https://<ZFPServerIP>/ZFP?mode=proxy#viewall&un=<user_name>&pw=<authentication_key>&sui=<study_search_parameters>”

Launching ZFP Using the Script from a Web-based Application

Some EMR or third-party application, even though being web-based, may not be able or willing to pre-initialize the ZFP viewer and load new studies in an existing ZFP instance. In this case, the VB script presented above can be invoked through a Web page, provided that the Internet Explorer settings allow it.

Besides the above steps, configure the EMR or third-party application to launch the following URL:

https://<ZFPServerIP>/ZFP/Scripts/OpenAPI/Trampoline.html?launchUrl=https%3A%2F%2F<ZFPServerIP>%2FZFP%2F%3Fmode%3Dproxy%23viewall%26sui%3D<study_search_parameters>%26un%3D$user_name%26pw%3D$authentication_key
Appendix H: Proxy Launch with Single ZFP (ZFP Configured Against EA and CPACS as DICOM Sources)

When ZFP is configured against EA and CPACS as DICOM sources, and ZFP is launched through proxy mode, the following is the expected behavior from ZFP:

- When ZFP is launched through a proxy URL, ZFP will search for primary studies against both CPACS and EA back ends.
- There are two set of search parameters that ZFP proxy launch supports. Historically only one set of parameters could be used depending on what backend you want to query. Now irrespective of which set of parameters you use ZFP will search both CPACS and EA.
- Please note that parameters from the different set cannot be intermixed.
- If some parameter is not valid against a particular backend it will be ignored. For e.g., if the url is passed like this
  
  http://localhost/zfp?mode=proxy&view&ris_pat_id=DMGRPAPH016&exam_stat_desc=Arrived
  
  then ZFP will search CPACS for studies belonging to patientId 'DMGRPAPH016' and which are in arrived status and it will query EA for studies belonging to patientId 'DMGRPAPH016', since exam status does not apply to EA backend.

  If we want to query a specific backend like CPACS only or/and specific EA archives we can do so by specifying the ae_title property.

- After the study search is complete, the search results will be displayed based on the proxy command that is used: showlist, view, or viewall.
  - showlist: Shows the study search page with the search results. The user can click on a primary study to be launched in the viewer. Prior studies for the patient will be fetched and shown in the Navigator from all the sources (CPACS and EA).
  - view: If the study search results in more than one study, then the study search page will be displayed with the searched results. The user can click on a primary study to be launched in the viewer. Prior studies for the patient will be fetched from all the archives/sources. If the study search results in a single study, then the study will be launched directly in the viewer, with priors displayed in the Navigator from all the sources.
  - viewall: The latest study from the search result will be loaded in the viewer as the primary study, and the rest of the studies will load in the Navigator as priors. In this mode, the priors are loaded from the initial search results, which means the priors appear from all sources, or specified sources passed in the ae_title property.
Appendix I: General Troubleshooting

When using Windows AD (NTLM) authentication, page flickering occurs when loading ZFP (with or without URL launch)
Complete the steps below.

Note: Use these steps during the below scenarios only:

- When launching ZFP in standalone mode with Windows AD (NTLM) authentication.
- When launching ZFP in proxy mode without ‘un’ and ‘pw’ parameters, that is, instead of these parameters, the current user’s login credentials are used for authentication.

1. Open IIS.
2. Select the ZFP web application.
3. Enable **Windows Authentication**.
4. Disable **Anonymous Authentication**.
5. Restart IIS.