Developers Guide

Sectra EPR Integration (URL) and External Thumbnail Viewer

Version 16.1, March 2014

SECTRA

About this document

This developer's guide is intended for developers of health care applications that are to integrate with the IDS7, LiteView or Order Management, using the EPR Integration (URL) interface or to integrate with the IDS7 or LiteView using the "External Thumbnail Viewer" interface.

The EPR Integration (URL) interface makes it possible for other applications to interact and send commands to IDS7, LiteView and Order Management controlling which information they should show. The External Thumbnail Viewer allows the integration of thumbnails in other applications and launching IDS7 or LiteView from these.

This guide contains tutorials and recommendations for how to integrate an application with any of the interfaces, as well as a reference specification of the interface.

Disclaimer

All information in this manual has been anonymized. Any resemblance of names, information or images in this manual to actual persons or events is coincidence. Sectra AB is not responsible for problems caused by changes in the operating characteristics of the computer hardware or operating system, which are made after the delivery of the software, or for problems that occur as a result of the use of Sectra software in conjunction with non-Sectra software other than the software explicitly covered in this documentation. For a complete description of the warranty, refer to the End User License Agreement supplied with the Sectra product.

Feedback

As part of our commitment to provide the highest quality products and services, we would like to encourage feedback on the quality of this documentation. If you think we can improve this document in any way, then please send your suggestions to info@sectra.se.

Contents

1 Introduction

The following topics are included in this chapter:

- EPR Integration (URL)
- External Thumbnail Viewer
- Related documents

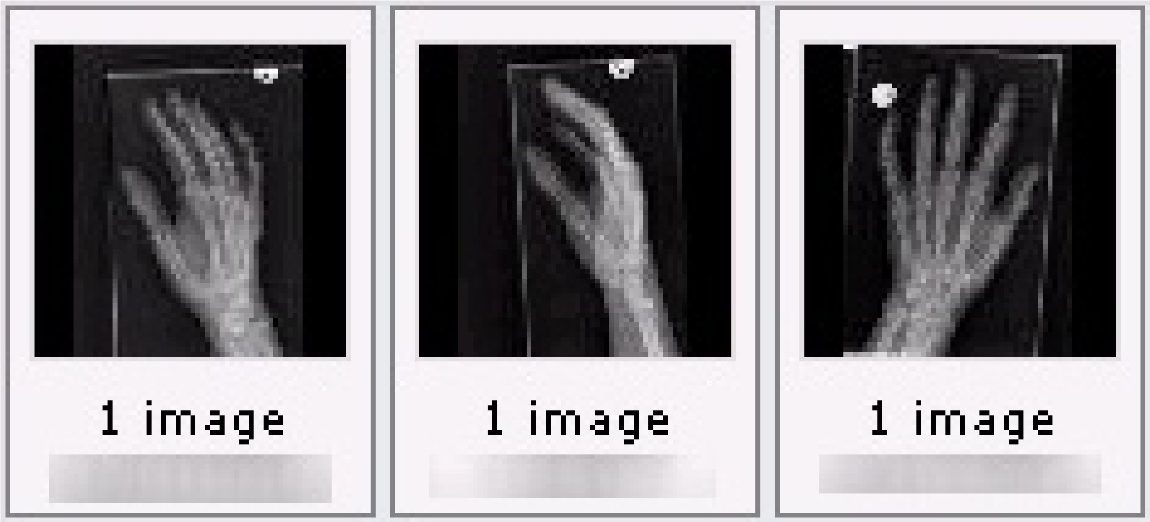
1.1 EPR Integration (URL)

This document describes how an external application can launch IDS7, LiteView or Order Management, using the "EPR Integration (URL)". IDS7 and LiteView can be launched to directly display examination data and images stored in Sectra PACS. Sectra Order Management can be launched in desired mode and state. The products can be launched without the user having to provide log on credentials. This provides a very efficient way for external applications - for example an EPR/HIS client - to also provide the users with examination data and images from the PACS. The products are launched by generating a specific web-address, a URL, while maintaining access security through the use of temporary access control keys.

This document is part of the Software Development Kit (SDK) that includes the items needed to implement such an integration with IDS7, LiteView or Order Management. It provides documentation as well as code examples to generate the URL required.

1.2 External Thumbnail Viewer

The External Thumbnail Viewer support in EPR Integration (URL) makes it possible to launch IDS7 or LiteView from an ASP.NET page embedded in an external application. The embedded ASP.NET page will show the thumbnail images of a specific examination and IDS7 or LiteView can be launched by clicking the thumbnail images.



Example of the thumbnail control

1.3 Related documents

[1] Sectra User's Guide IDS7

[2] User's Guide LiteView

[3] System Administrator's Guide Sectra Healthcare System

[4] Conceptual Overview Sectra PACS components and how they interact

[5] Sectra PACS Product Catalog

2 Getting started

The following topics are included in this chapter:

- Requirements for EPR Integration (URL)
- Requirements for External Thumbnail Viewer
- Launching IDS7 or LiteView using EPR Integration (URL) for the first time
- Launching Order Management using EPR Integration (URL) for the first time
- Using automatic log on
- Using automatic creation of user accounts
- Using external roles
- Launching IDS7 or LiteView from the External Thumbnail Viewer for the first time

This section describes how to get started with the EPR Integration (URL) and External Thumbnail Viewer.

See also

- For an introduction to EPR Integration (URL), see chapter 3 About EPR Integration (URL).
- For information about backwards compatibility with IDS5/web, see section 3.5 About backwards compatibility.

2.1 Requirements for EPR Integration (URL)

This section lists the requirements that must be fulfilled for the EPR Integration (URL) to work with IDS7, LiteView or Order Management:

IDS7 requirements

- **System requirements** - The computer where IDS7 is launched must meet the system requirements for IDS7. See IDS7 documentation for details

Note: Due to security settings in Internet Explorer, the use of EPR Integration (URL) may result in a blue background screen instead of starting IDS7. Please see tech note TN-JSJG-8NNJYZ for more information about this problem.

- **Licenses** - There must be a basic product license for IDS7 available. Either connected to the workstation, or connected to the PACS role/user used. A license for the IDS7 add-in “EPR Integration (URL)” is also required.
- **User account** - The PACS user account used must either exist in WISE domain or AD domain. See section 3.3 About user handling.

- **Permissions** - The PACS user account must have certain permissions set to be able to launch IDS7. See section 3.3 About user handling.
- **The URL** - The URL used must be a valid URL, see chapter 4 About the URL for EPR Integration (URL).

LiteView requirements

- **System requirements** - For LiteView system requirements, refer to System Administrator's Guide Sectra Healthcare System [3]
- **Licenses** - There must be a basic product license for LiteView available. A license for the IDS7 add-in “EPR Integration (URL)” is also required.
- **User account** - The PACS user account used must either exist in WISE domain or AD domain. See section 3.3 About user handling.
- **Permissions** - The PACS user account must have the permission to start LiteView. See section 3.3 About user handling.
- **The URL** - The URL used must be a valid URL, see chapter 4 About the URL for EPR Integration (URL).

Order Management requirements

- **System requirements** - The computer where Order Management is launched must meet the system requirements for Order Management. See Order Management documentation for details.
- **Licenses** - The EPR Integration (URL) license must be enabled in Sectra RIS, alongside with at least one of the main licenses Order Entry or Appointment Booking.
- **User account** - The user account used must be available in Sectra Enterprise Manager. See section 3.3 About user handling.
- **Permissions** - The user account must have certain permissions set to be able to launch Order Management. See section 3.3 About user handling.
- **The URL** - The URL used must be a valid URL, see chapter 4 About the URL for EPR Integration (URL).

2.2 Requirements for External Thumbnail Viewer

In addition to the requirements for EPR Integration (URL), the following requirements must be fulfilled for the External Thumbnail Viewer to work:

- **Web browser** - Internet Explorer 7, 8 and 9 are supported.
- **EPR Integration (URL) configuration** - EPR Integration (URL) needs to be configured correctly.
- **Licenses** - A External Thumbnail Viewer license for the Sectra Healthcare Server is required.
- **The URL** - The URL used must be a valid URL, see chapter 7 About the URL for External Thumbnail Viewer.

2.3 Launching IDS7 or LiteView using EPR Integration (URL) for the first time

1. **Check requirements**

   Make sure that the requirements for the EPR Integration (URL) are met. See section 2.1 Requirements for EPR Integration (URL).

2. **Add licenses**

   Use the **Licenses tool** in WISE/tools to add required licenses. See section 2.1 Requirements for EPR Integration (URL).

3. **Create a new PACS role**

   Log on to Sectra Enterprise Manager and click **New PACS role** to create a new PACS role for the EPR Integration (URL). Name it “PACSEprRole”.

4. **Connect a basic product license (IDS7 only)**

   In Sectra Enterprise Manager, click **Properties** for the new role "PACSEprRole" and select the tab **Licenses**. Connect the role to an IDS7 basic product license by using the drop-down list.

   This requires that either the Concurrent User Engine (WISE add-in) or an IDS7/cx basic product license is available. If neither of them are available, then use the **PACS Infrastructure** tool to connect the IDS7 basic product license to the workstation instead.

   Using a floating license (as provided with Concurrent User Engine or IDS7/cx licenses) for the EPR Integration (URL) is highly recommended since it will make the integration very flexible. IDS7 can then be launched from any workstation (known or unknown). Otherwise the workstation (computer) must be pre-defined in the PACS infrastructure in WISE/tools, and an IDS7 basic product license be bound to the workstation.

5. **Create a new PACS user for testing purposes**

   In Sectra Enterprise Manager, click **New PACS user** to create a new PACS user to be used when testing the EPR Integration (URL). Name the user “EprTestUser”. Make it a member of the PACS role “PACSEprRole”.

   This PACS user account should be removed when the integration is completed.

6. **Set role permissions and properties**

   Use the Sectra Enterprise Manager to set the permissions for the PACS role “PACSEprRole”. See section 2.1 Requirements for EPR Integration (URL).

   For IDS7, set also the **Preferred workspace** to IDS7.

7. **Launch the application through the EPR Integration (URL) web page.**

   Open a web-browser and give it a URL, where `<hostname>` should be replaced with the network name of the Sectra Healthcare Server. For IDS7, use the following URL:

   ```
   http://<hostname>/IDS7/3pstart.aspx
   ```

   For LiteView use the following URL:

   ```
   http://<hostname>/LiteView/index.html
   ```

This will launch the application and display the log on window. Manually enter the user name “EprTestUser” and supply the password. Verify that the log on works.

See also

- System Administrator's Guide Sectra Healthcare System [3] for information about IDS7 Starter, a program that can speed up the start of IDS7 with EPR Integration (URL).

2.4 Launching Order Management using EPR Integration (URL) for the first time

1. **Check requirements**

   Make sure that the requirements for the EPR Integration (URL) are met. See section 2.1 Requirements for EPR Integration (URL).

2. **Add licenses**

   Request a Sectra technician to enable the EPR Integration (URL) option in Sectra RIS (Europe/Pacific) and at least one of the main licenses Order Entry or Appointment Booking.

3. **Create a new role for testing purposes**

   Use Sectra Enterprise Manager to create a new role to be used when testing EPR Integration (URL). Name it “OrderManagementEprRole”.

   This role should be removed when the integration is completed.

4. **Set role permissions and properties**

   Use Sectra Enterprise Manager to set the permissions for the role “OrderManagementEprRole”. The role must have the permission **Start Sectra Order Management** (at least).

5. **Create a new user for testing purposes**

   Use Sectra Enterprise Manager to create a new user to be used when testing the EPR Integration (URL). Name the user “OrderManagementEprTestUser”. Make it a member of the role “OrderManagementEprRole”.

   This user account should be removed when the integration is completed.

6. **Launch Order Management in a browser**

   Open a web-browser and give it the following URL, where hostname should be replaced with the network name of the Sectra Workflow Server:

   ```
   https://<hostname>/OrderManagement/3pstart.aspx
   ```

   This will launch Order Management and display the log on window. Manually enter the user name “OrderManagementEprTestUser” and supply the password. Verify that the log on works.

2.5 Using automatic log on

If the external application launching IDS7, LiteView or Order Management via EPR Integration (URL) can be trusted, it can be granted permission to bypass the normal log on procedure. The responsibility of authenticating the user is then handed over to the external application. By appending log on credentials to the URL, the external application will be able to launch the applications without user interaction, and directly display the wanted data.

Warning: When the automatic log on is used, the user is only identified by the `user_id` URL parameter. No user authentication will be made by the launched product. Only the launching system is authenticated as a trusted system. For that reason it is of great importance that the user registries of the launching and the launched system are synchronized at all times. Mismatches between user_id in the launching and launched system might result in unauthorized access to medical records.

The following is required for automatic log on to work:

- The `user_id` must be specified in the URL. See also section 4.1 URL parameter descriptions.
- The user account specified by `user_id` must exist in the specified domain. If the user account does not exist the log on window is displayed. See also section 2.6 Using automatic creation of user accounts.
- The specified user account must have permission to start the product to be launched. See section 3.3 About user handling.
- To bypass the log on procedure a *system password* is required. The same system password in is set in both the Sectra Healthcare System as well as in the external application. See section 3.2 About the system password.

2.6 Using automatic creation of user accounts

A very useful feature of the EPR Integration (URL) is the possibility to automatically create user accounts for users accessing the Sectra PACS for the first time. That is, when the URL contains a user ID that does not exist in WISE, a new PACS user account is created instead of rejecting the user. This feature is only available for PACS users.

There are configurations for how the new account should be created (user name prefix, role membership and password policy). If an IDS7 or LiteView basic product license is connected to the role, then the new user will be able to launch IDS7 or LiteView and get started, without any password confirmation or system administrator interaction.

Note: This feature is not recommended for Order Management due to the vast amount of users it would create in WISE.

▶ **To automatically create a new user account for new users**

This procedure is the continuation of section 2.5 Using automatic log on.

1. **Open Url Integration configuration tab**

   In Sectra Enterprise Manager, under **Servers**, select the local WISE server.

   Click **Properties**, select the tab **Url Integration**.

2. **Enable automatic creation of users**

   In the tab Url Integration, do the following:

   - Select the **Automatically create users**.
   - Enter a prefix in **Default username prefix**, e.g. “UrlAuto_”. This will be the prefix of all automatically created PACS user accounts.

Tip: The automatically created users can be numerous. To make them less conspicuous in lists of user accounts (like in Sectra Enterprise Manager) consider using a prefix that begins with “ZZ” or similar. This will put them last in alphabetically sorted lists.

   - In the list of **Roles for automatically created users**, add the role “PACSEprRole” and set it as the preferred role.

   Do *not* select the **Use default password** option, unless required by the circumstances. If not selected, an automatically generated secure password will be assigned to the new user account. This works since the user is created for log on via the URL integration that bypasses the normal password check.

Example: Launching IDS7 via EPR Integration (URL) with `user_id=NewUser1`, will automatically create the new PACS user account “UrlAuto_NewUser1” in WISE and automatically log on to IDS7 as the user “UrlAuto_NewUser1”, without displaying the log on window.

```
http://<hostname>/IDS7/3pstart.aspx?user_id=NewUser1&time=<time>&key=<key>
```

2.7 Using external roles

The external system launching IDS7 via EPR Integration (URL), may attach an additional identifier in the URL called the `HIS id` . In the Sectra Healthcare System, this `HIS id` is mapped to a PACS role or an SHS role. Using a `HIS id` will specify the preferred role for the user logging on, i.e. it will temporarily override the preferred role setting that normally applies to the user.

This makes it possible to easily define custom permissions and settings for groups of users that are logged on through the EPR Integration (URL) without the need to synchronize role membership between the external system and Sectra Healthcare System. All configurations such as licenses, access permissions etc. will be the ones of the temporary preferred role. The temporary preferred role will only last for the current session, until the user logs out.

The `HIS id` is a string, defined in the external system. The mapping between `HIS id` and the corresponding preferred role in the Sectra Healthcare System is configured in the **Role** properties dialog in the Sectra Enterprise Manager.

Note: This feature is not supported by Order Management.

Note: In order to automatically change preferred role for users that are logged on using EPR Integration (URL), the role permission “Allow auto-adding url users” is required.

▶ **To map external roles to roles defined in Sectra Healthcare System**

1. **Open Role configuration tab**

   In Sectra Enterprise Manager, select **User Administration**->**Roles**.

   Double-click the role that you want to map to an external role or group, and select the tab **General**.

2. **Set the external role id**

   In the tab General, enter the identifier (the `HIS id` string) for the external role in the **HIS ID** field.

3. **Change permissions to allow adding users to the role**

   Select the tab **Permissions**.

   Set the permission **Allow auto-adding url users**.

   Click **OK**.

Example: Launching IDS7 via EPR Integration (URL) with `user_id=IDS7EprTestUser&his_id=HisRoleId`, will automatically log on the user "IDS7EprTestUser" and temporarily change its preferred role to the role for which the HIS id is set to "HisRoleId".

```
http://<hostname>/IDS7/3pstart.aspx?user_id=IDS7EprTestUser
 &his_id=HisRoleId&time=<time>&key=<key>
```

2.8 Launching IDS7 or LiteView from the External Thumbnail Viewer for the first time

This is a brief description of how to get started with the External Thumbnail Viewer and how to launch IDS7 or LiteView from the External Thumbnail Viewer for the first time.

▶ Get started with the External Thumbnail Viewer

1. **Check requirements**

   Make sure that the requirements for the EPR Integration (URL) and External Thumbnail Viewer are met. See section 2.1 Requirements for EPR Integration (URL).

2. **EPR Integration (URL) configuration**

   Make sure that EPR Integration (URL) is configured and is working.

3. **Add license**

   Use the **Licenses tool** in WISE/tools to add the license for the External Thumbnail Viewer. The External Thumbnail Viewer license starts with SHS-4-0-SITE.... The presentation string for the External Thumbnail Viewer license in WISE/tools is **SHS #External Thumbnail Viewer 12.x SITE Lic. ...**

4. **Start ViewerExternalThumbnails.exe**

   Copy the content of the bin folder (from the SDK) to your local workstation and start the application by double-clicking on `ViewerExternalThumbnails.exe`. For more information about ViewerExternalThumbnails.exe see section 6.4 About ViewerExternalThumbnails.exe. Enter the information according to the list in section 6.4 About ViewerExternalThumbnails.exe.

5. **Generate a URL**

   Press **Step1. Generate URL** on the `Get thumbnails` tab to process the information and generate a URL.

6. **View the thumbnails**

   Press **Step2. Get thumbnails** to view the thumbnail images in the window.

7. **Launch IDS7 or LiteView from the thumbnail view**

   Click on one of the thumbnail to launch IDS7 or LiteView. This requires that the option 'Launch on click' is 'Yes'.

3 About EPR Integration (URL)

The following topics are included in this chapter:

- Url Integration properties in Sectra Enterprise Manager
- About the system password
- About user handling
- About time limited URL
- About backwards compatibility
- About integrating Sectra Order Management with a HIS

The IDS7 workstation is a web-based application, meaning that it is started by clicking a start link on a web page. The start link (URL) points out a web-service part of a Sectra Healthcare System (Sectra Healthcare Server or Sectra Satellite Server). When launched, the IDS7 application is automatically downloaded from the server (using Microsoft ClickOnce technology) and started displaying a log on window.

LiteView and Order Management are true web-applications. They are only used through a web-browser.

Beside the normal log on procedure for these products, the Sectra Healthcare System also provides a way for a trusted external system to launch the products from the external application by providing a specific URL. This is called EPR Integration (URL) and is described in this SDK.

The URL can be compiled in different ways to control the behavior of the launched application. For example; if the URL only contains data to identify the current user (no patient or examination identifiers), the user will be automatically logged on and the user can then search for any patient within the system. If the URL also contains patient identifiers, an automatic search for patient information will be performed.

See also

- chapter 2 Getting started
- chapter 4 About the URL for EPR Integration (URL)
- chapter 5 How to compile the URL for EPR Integration (URL)

3.1 Url Integration properties in Sectra Enterprise Manager

Properties for EPR Integration (URL) are set in Sectra Enterprise Manager. The configuration differs dependent on the domain authenticating the user.

▶ **To open the Url Integration properties window**

1. Log on to Sectra Enterprise Manager
2. Select the server type in the tree view:

   For PACS users and AD users select **Servers->WISE Servers**.

   For SHS users select **Servers->Sectra Healthcare Servers**.
3. Double-click the local WISE server, or the local Sectra Healthcare Server, in the list to open the **Properties** window.
4. In the **Properties** window, select the tab **Url Integration**. The following settings are available:

   System password
   An encryption key shared between the launching and launched systems. See section 2.5 Using automatic log on.

   Max allowed age of access key
   Sets the time during which generated URLs are considered valid. See section 3.4 About time limited URL.

   Automatically create users
   This setting is available for the WISE server only and applies to PACS users.

   When enabled, PACS users are automatically created during log on, see section 2.6 Using automatic creation of user accounts.

3.2 About the system password

The system password is an encryption key shared between the launching and launched systems, like a shared secret. When both systems have the same system password set, it is possible to use automatic log on via EPR Integration (URL). See section 2.5 Using automatic log on.

The system password is used as an encryption key when encrypting and decrypting the URL. Only URLs based on the correct system password will be processed. This way the password itself is not transmitted.

Important: The external application must handle the system password in a secure way, making any unauthorized access impossible. The password must never be disclosed to users of the external application.

Note: Using a system password will only bypass the authentication (the password security check). Each log on will, as usual, be carried out using the user's normal user account. See also section 3.3 About user handling.

The system password can be set in two locations, depending on the type of user launching the application via EPR Integration (URL):

- For PACS users and AD users the system password is set for the local WISE server.
- For SHS users the system password is set for the local Sectra Healthcare Server.

The system password is set from the **Url Integration** properties window, see section 3.1 Url Integration properties in Sectra Enterprise Manager.

In case users of all types may launch the application via EPR Integration (URL), it is recommended to set the same system password in both locations.

3.3 About user handling

To be able to log on to the IDS7, LiteView or Order Management a user account is always required. This is also true when logging on using the EPR Integration (URL). Which user account to be used is defined by the URL parameter `user_id`.

The user_id is defined on the format "`<domain>\<user>`", where the <domain> specifies a local WISE domain, AD domain or SHS domain. If the <domain> is omitted, WISE domain is assumed. Specifying domain is required when logging on as an AD user or an SHS user.

The <domain> is the domain identifier shown in the Sectra Enterprise Manager.

▶ To view the domain identifier

1. In Sectra Enterprise Manager, select **User Administration**->**Domains**.
2. In the list, double-click the local domain.
3. In the **Domain Properties** window for the domain, the **Domain identifier** is listed. Examples:

WISE domain	MyWiseHost_7800
AD domain	Sectra.com
SHS domain	5ff31a12-5129-e011-852f-001560a767d4

Table 3.1 Authorization of users connecting via EPR Integration (URL)

EPR Integration (URL) feature	Authorization required
To launch IDS7	● Role permission "Start IDS7 Application" ● IDS7 license
To launch LiteView	● Role permission "Start Sectra LiteView" ● LiteView license
To launch Order Management	● Role permission "Start Sectra Order Management" and at least one of the permissions for creating requests or scheduling examinations. ● Order Entry license and/or Appointment Booking license.
To automatically change preferred role for users that are logged on using EPR Integration (URL)	● Role permission "Allow auto-adding url users"

See also

- Parameter `user_id` in section 4.1 URL parameter descriptions

- section 2.5 Using automatic log on
- section 2.6 Using automatic creation of user accounts
- section 2.7 Using external roles on automatically changed preferred role

3.4 About time limited URL

The URL contains a time stamp to represent its creation. When processing the URL in the Sectra Healthcare System, the current time is compared to the time stamp of the URL. This is needed to help ensure log on credibility for the external application.

The URL validity time is configurable in the **Url Integration** configuration in Sectra Enterprise Manager. This way it is possible to set the time span during which a URL is valid. It also induces a certain time synchronization dependency between the external application and the Sectra Healthcare System. In a scenario where the URL validity time is smaller than the system clock UTC difference between the external application and Sectra Healthcare System, a URL sent between the two will not be valid. Since the time stamp is based on UTC, it works also when spanning several time zones.

See also

- Parameter `time` in section 4.1 URL parameter descriptions

3.5 About backwards compatibility

URL integration was first introduced with IDS5/web. As the IDS5/web product is discontinued, the same integration is now available with IDS7 or LiteView. When used in conjunction with IDS7 or LiteView it is called EPR Integration (URL).

The URL syntax used for IDS7 and LiteView is not compatible with the IDS5/web syntax. The encryption method used to secure EPR Integration (URL) has been improved, which breaks backwards compatibility.

Over the past, there have been utilities offered to help generating the URL, i.e. `viewer_3p_launch.exe` for IDS5/web and `Viewer3pLaunch.exe` for previous versions of IDS7. This is not the case anymore, and new applications must implement the URL generation on their own, see chapter 5 How to compile the URL for EPR Integration (URL)

This also means that an application developed to create URLs for IDS5/web needs to be replaced in order for URL integration to work together with IDS7 or LiteView.

Note: When replacing IDS5/web, please verify that the workstations where the URL integration is used also meet the system requirements for IDS7 or LiteView.

In version 12.1.1, the encryption method used to secure EPR Integration (URL) was modified in an effort to move towards standard algorithms. Applications with support for EPR Integration (URL) added prior to 12.1.1 need to be updated accordingly.

3.6 About integrating Sectra Order Management with a HIS

When integrating Order Management with a HIS, the integration can be made to behave differently if an embedded web control is used compared to a standalone web browser. Embedding Order Management is recommended if patient synchronization with another program is used.

When a HIS launches Order Management with some patient, and the user then selects another request or another patient in the HIS, Order Management will not follow that patient change, instead the first patient will still be selected. The HIS can issue new EPR URLs to signal Order Management to change the request or the patient, but only if Order Management is running in a web control hosted within the HIS. If a standalone browser is used to host Order Management, this EPR launch from the HIS will only launch new instances of Order Management (new web browser tabs or browser windows). Existing tabs or windows from previously launched EPR URLs will not be closed automatically, however for the following scenarios an automatic redirect to an information page will occur:

- If a new EPR URL is launched with parameter `stop` an automatic redirect and a forced logout will occur in all existing tabs or windows running Order Management.
- If a new EPR URL is launched with a different user (i.e. value of parameter `user_id` was changed) an automatic redirect and a forced logout of previous user will occur in all existing tabs or windows running Order Management.
- If a new EPR URL is issued with a different patient (i.e. value of parameter `pat_id` was changed) and changing patient is prohibited in Order Management (i.e. parameter `allow_pat_change` has value 0) then an automatic redirect will occur in all existing tabs or windows running Order Management.
- If a new EPR URL is launched with changed value for parameter `allow_pat_change` an automatic redirect will occur in all existing tabs or windows running Order Management.

Warning: When using patient synchronization with another program the following should be considered if not embedding Order Management.

- Automatic redirects and forced logouts in existing tabs or windows due to new launches of EPR URLs will discard any unsaved changes such as a request not submitted or a booking not completed.
- It is not possible to run Order Management in a stand alone web browser with different users for different tabs or windows.

4 About the URL for EPR Integration (URL)

The following topics are included in this chapter:

- URL parameter descriptions
- URL parameter encoding
- Login to IDS7
- URL examples
- Transfer protocol

The URL is a compiled string that specifies how IDS7, LiteView or Order Management should be launched, and what information they should display.

These are the main parts of the URL:

```
<protocol>://<hostname>/<webservice>?<parameters>
```

<protocol> http or https depending on the configuration of the web-server.

<hostname> The hostname of the Sectra Workflow Server, Sectra Healthcare Server (or the Sectra Satellite Server) from where IDS7, LiteView or Order Management will be launched.

<webservice> The path to the web service of Sectra Healthcare Server (or Sectra Satellite Server), Sectra LiteView Server or Sectra Workflow Server. The default path to the products implementing EPR Integration (URL) will launch the default instance and version of the web service. The default paths are:

IDS7 has the default webservice name: “IDS7/3pstart.aspx”

LiteView has the default webservice name: "LiteView/index.html#launch"

Order Management has the default webservice name: "OrderManagement/3pstart.aspx"

<parameters> Several parameters can be passed on to the launched product to control how and what it should display. See section 4.1 URL parameter descriptions.

See also

- section 4.4 URL examples

4.1 URL parameter descriptions

Table 4.1, “URL Parameter description” lists the parameters of the URLs and their meaning. The parameters supplied must be encoded as specified in section 4.2 URL parameter encoding.

Table 4.1 URL Parameter description

Parameter	IDS7	LiteView	Order Management	Description
`stop`	X	X	X	The value of the `stop` parameter should, when used, always be "1". A URL that contains the `stop` parameter will cause any running instance of IDS7 on the current client to terminate. No new instance of IDS7 will be started, even if the URL contains login information or examination identifiers. When used with LiteView or Order Management, it ends the current session and forces logout of the current user. It can take a few seconds before the client windows act according to this. No new server requests will be successfully authorized in the meantime, but any information still visible in the browser, can be read by a user until the forced logout. If further url-commands are issued before these seconds have passed the user might not be logged out and will still be able to use the system. For LiteView, the `stop` parameter will also terminate any running IDS7 started by the LiteView client. The "stop" command can be used without any other parameter specified. **Note:** *Also see TN-JSJG-8NNJYZ regarding stopping IDS7 from LiteView.*
`user_id`	X	X	X	The user name to be used to login to the launched product. If the " `user_id` " parameter is omitted, the launched product will start with requesting the user to log on with user name and password before it displays the specified examinations. It can optionally be formatted to target a specific `domain` by separating the `domain_id` and `user_id` by a backslash. Example: " `<domain_id>\<user_id>` " See also section 3.3 About user handling.
`time`	X	X	X	Timestamp for the creation of the URL. The timestamp is the system clock time measured in seconds since January 1st 1970 (see ANSI-C " `time()` " function). See also section 3.4 About time limited URL. **Note:** *The timestamp should not be compensated for daylight saving.*
`key`	X	X	X	Access control key, represented as a hexadecimal string. The access control key is based on the `time` timestamp and it is only valid for a limited period of time. See section 5.1 Access control key creation.
`pat_id`	X	X	X	Patient identifier string.
`acc_no`	X	X	X	Accession number string. The parameter value can be a list of accession numbers separated by character "^" if several accessions are to be viewed. All accession numbers must be associated with the same patient. When launching Order Management to schedule a request defining this or `ext_acc_no` is required. If used it must only be one accession number.

Parameter	IDS7	LiteView	Order Management	Description
exam_id	X	X		Examination identifier string. The parameter value can be a list of examination identifiers, separated by character "^" if several examination are to be viewed. **Note:** *The number of items in the list should be the same as the number of items in the acc_no list and both lists should the sorted so accession numbers and examination identifiers appears in pairs.*
mrn_integration_id	X	X		Sectra PACS can be configured to request the search result only from WISE servers belonging to a specific MRN group in the Sectra Healthcare System. This parameter can also be configured as a default value for each workstation. **Note:** *The parameter value is the MRN integration ID specified for the MRN group, and not the name of the MRN group.*
acc_no_group	X	X		Sectra PACS can be configured to request the search result only from WISE servers with a specific Accession number group ID in the Sectra Healthcare System. This parameter can also be configured as a default value for each workstation.
sop_uid	X	X		Specifies a DICOM SOP Instance UID. This parameter must be used in combination with one single accession number and exam id. IDS7 will create a presentation displaying the specified image if the UID is unique within the specified exam.
frame_number	X	X		Can be used in combination with DICOM SOP Instance UID to specify which frame to display in a multi-frame image. Frame numbers start at 1. This parameter is optional (default value is 1).
pat_name	X			Patient name that is displayed in the Picture and Video import dialog in IDS7 if a patient matching the patient identifier (pat_id) does not already exist in PACS. The patient name should be in a valid DICOM person name format, including the '^' characters between the name parts.
pat_birth	X			Patient birthdate that is displayed in the Picture and Video import dialog in IDS7 if a patient matching the patient identifier (pat_id) does not already exist in PACS. The patient birthdate should be in a valid DICOM date format.
pat_sex	X			Patient sex that is displayed in the Picture and Video import dialog in IDS7 if a patient matching the patient identifier (pat_id) does not already exist in PACS. The patient sex should be in a valid DICOM format, EMPTY, M, F or O (Empty, Male, Female or Other).
ref_unit	X		X	Referring unit that is displayed in the Picture and Video import dialog in IDS7 if a patient matching the patient identifier (pat_id) does not already exist in PACS. In Order Management specified parameter value is expected to match External Id for a referring unit. The referring unit will be used by Order Management to determine which timeslots the user will have access to view when using the scheduler_overview-command.

Parameter	IDS7	LiteView	Order Management	Description
`his_id`	X			Identifier for an external role or group that is matched against a role in the Sectra Healthcare System. When a user is logged in its preferred role will temporarily be set to the role that matches the HIS-id. This makes it possible to for example define special permissions or settings for groups of people defined by their id in the external system, i.e. the HIS-id. The mapping between HIS-id and the corresponding role in the Sectra Healthcare System is configured in the **Role** properties dialog in the Sectra Enterprise Manager.
`close_popup`	X			The value of the `close_popup` parameter should, when used, always be "1". If the URL contains the `close_popup` parameter, an attempt will be made to close the web browser window that the URL is opened in. This is useful when the external system is a standalone application (or web-application) and a new browser session or window appears each time IDS7 is started. The `close_popup` parameter is however not suitable if the external application is a web-application and opens the URL inside its own browser window. **Note:** *The `close_popup` parameter should not be included in the hash used to generate the access control key. This makes it possible to add it as a conditional extra parameter at a separate stage of the integration.*
`init`	X			The purpose of the `init` parameter is to initialize an instance of IDS7 in preparation for subsequent queries. The command is ignored if IDS7 is already running. The `init` parameter is only meaningful when quick launch is configured for the workstation group. When configured, IDS7 will start, show an empty presentation, and then log off and run in the background. If not configured, IDS7 will exit immediately after starting. The value of the `init` parameter should, when used, always be "1". The "`init`" command can be used without any other parameter specified. See System Administrator's Guide Sectra Healthcare System [3] for information about quick launch and running in the background.

Parameter	IDS7	LiteView	Order Management	Description
`cmd`			X	Specified parameter value determines which command that will be performed by Order Management. Possible values for the parameter are the following values: ● `create_request` - creation of requests for the given patient (additional parameters required: `pat_id`) ● `schedule_request` - scheduling of an already created request that has to exist in the RIS database (additional parameters required: `pat_id` and `ext_acc_no/acc_no`) ● `scheduler_overview` - shows an overview for the given referring unit (additional parameters required: `ref_unit`) ● `search_patient` - shows an overview for the given patient (additional required parameters: `pat_id`, additional optional parameters: `acc_no/ext_acc_no`). With the optional parameter, the given request will be shown. ● `order_overview` - shows the Order Overview
`ids7_cmds`	X			Specified parameter value determines which command that will be performed by IDS7. The parameter value can be a list of commands separated by character "^" if several commands are to be executed. Possible values for the parameter are the following values: ● `open_ipv_window` - open the Import Pictures and Videos window **Note:** *When using the open_ipv_window command, the **Image Window** will also be shown by default. This can be disabled in **Advanced Workstation Configuration** by disabling the **Automatically open image windows** setting.*
`ext_acc_no`			X	The external id of a request that has been sent to RIS. Defining this or `acc_no` is required to schedule a request.
`date_and_time`			In future release	The date and time to initially show in the scheduler. It should be on the given format: yyyyMMddHHmmss (hours must be given on 24-hour format).
`allow_pat_change`	X	X	X	Defaults to `1` which is enabled but if set to `0` (disabled) the user will not be able to change patient within IDS7, LiteView or Order Management. **Note:** *If `allow_pat_change` is set to 0 and Order Management is integrated in a standalone web browser any existing tabs or windows running Order Management will get automatically redirected and any unsaved changes will be discarded if a new EPR URL launch is issued with a changed value for parameter `pat_id`.*

4.2 URL parameter encoding

The parameter values of the URL are only allowed to contain characters listed in Table 4.2, "Allowed characters in parameter values". All other characters must be substituted with an escape sequence.

Table 4.2 Allowed characters in parameter values

Character	Description
A-Z	Upper case characters
a-z	Lower case characters
0-9	Digits
.	Dot
-	Dash
_	Underscore
!	Exclamation mark
(	Left-hand parenthesis
)	Right-hand parenthesis

The escape sequences are character triplets, consisting of the percent character "%" followed by two hexadecimal digits representing the ASCII code of the substituted character. For example, the back-slash character "\" is substituted with "%5C".

Note: Character "^" should not be substituted when it is used as a list separator.

4.3 Login to IDS7

If the full set of `user_id`, `time` and `key` parameters are included in the URL, the user will be automatically logged in without having to enter the user name or password. IDS7 can also be launched with one or more of these parameters missing from the URL, in which case the user will have to manually login with user name and password. After a successful login, the application will display examinations according to the rest of the URL parameters.

If IDS7 is already running, no manual login with user name and password will be required for subsequent launches if the `user_id` parameter in the URL matches the currently logged in user. This can be used by the external system to direct the running IDS7 to new patients and examinations without using the system password. (Access control is maintained, since no automatic login will be performed in this scenario. The EPR integration only performs operations that the user could have done anyway while being logged in, for example by using the search feature in IDS7.)

Note: When using the `user_id` parameter in this way, neither the domain nor the `his_id` parameter will be considered when deciding whether the current session can be re-used or if a new login is required.

4.4 URL examples

The following sections contains URL examples for launching IDS7 and Order Management.

The examples applies to launching IDS7, LiteView or Order Management. For a description of the URL parts <protocol>, <hostname> and <webservice>, see chapter 4 About the URL for EPR Integration (URL).

To launch with automatic login and let the user manually browse the PACS or Order Management:

```
<protocol>://<hostname>/<webservice>?user_id=<user>&time=<time>&key=<key>
```

To launch with automatic login as a domain user:

```
<protocol>://<hostname>/<webservice>?user_id=<domain_id>\<user>&time=<time>&key=<key>
```

To view all examinations for given patient:

```
<protocol>://<hostname>/<webservice>?user_id=<user>&time=<time>&key=<key>
 &pat_id=<pid>
```

To view all examinations for a given request:

```
<protocol>://<hostname>/<webservice>?user_id=<user>&time=<time>&key=<key>
 &pat_id=<pid>&acc_no=<accno>
```

To view one specified examination:

```
<protocol>://<hostname>/<webservice>?user_id=<user>&time=<time>&key=<key>
 &pat_id=<pid>&acc_no=<accno>&exam_id=<eid>
```

To view all examinations for a selected set of requests for a given patient:

```
<protocol>://<hostname>/<webservice>?user_id=<user>&time=<time>&key=<key>
 &pat_id=<pid>&acc_no=<accno_1^accno_2^accno_3>
```

To view a selected set of examinations for a given patient:

```
<protocol>://<hostname>/<webservice>?user_id=<user>&time=<time>&key=<key>
 &pat_id=<pid>&acc_no=<accno_1^accno_2^accno_3>&exam_id=<eid_1^eid_2^eid_3>
```

To view a specific image:

```
<protocol>://<hostname>/<webservice>?user_id=<user>&time=<time>&key=<key>
 &pat_id=<pid>&acc_no=<accno>&exam_id=<eid>&sop_uid=<sop instance uid>
```

To view a specific frame in a multi-frame image:

```
<protocol>://<hostname>/<webservice>?user_id=<user>&time=<time>&key=<key>
 &pat_id=<pid>&acc_no=<accno>&exam_id=<eid>&sop_uid=<sop instance uid>
 &frame_number=<frame number>
```

To stop the running instance of on the current client:

```
<protocol>://<hostname>/<webservice>?stop=1
```

4.4.1 Order Management specific examples

To create a request for given patient using Order Management:

```
https://<hostname>/OrderManagement/3pstart.aspx?user_id=<user>&time=<time>
 &key=<key>&cmd=create_request&pat_id=<pid>
```

To schedule a submitted request using Order Management:

```
https://<hostname>/OrderManagement/3pstart.aspx?user_id=<user>&time=<time>
 &key=<key>&cmd=schedule_request&ext_acc_no=<extaccno>&pat_id=<patid>
```

Note: The request to be scheduled must exist in the Sectra RIS Database before it is possible to schedule it.

To access Diary Overview using Order Management:

```
https://<hostname>/OrderManagement/3pstart.aspx?user_id=<user>&time=<time>
 &key=<key>&cmd=scheduler_overview&ref_unit=<refUnitExtId>
```

To search for a patient using Order Management:

```
https://<hostname>/OrderManagement/3pstart.aspx?user_id=<user>&time=<time>
 &key=<key>&cmd=search_patient&ext_acc_no=<extaccno>&pat_id=<patid>
```

To open Order Overview Order Management:

```
https://<hostname>/OrderManagement/3pstart.aspx?user_id=<user>&time=<time>
 &key=<key>&cmd=order_overview
```

4.5 Transfer protocol

The examples in this document simulate the URL integration by pasting a URL into the address field of a web-browser. This is not the recommended way for URL Launch implementation. The implementation should be made at a lower level. The exact solution will differ depending on the platform hosting the implementation and the desired behavior of the implementation. For a lower level implementation of the URL Launch interface the transport protocol will have to be defined.

The following table shows the supported transport protocol for IDS7, LiteView and Order Management:

Table 4.3 Transport Protocol details

Product	Transport Protocol	Request Method
IDS7	HTTP or HTTPS	POST or GET
LiteView	HTTPS	GET
Order Management	HTTPS	POST or GET

The POST Request Method is recommended in front of GET since it will avoid leaving a trace in the browser history.

5 How to compile the URL for EPR Integration (URL)

The following topics are included in this chapter:

- Access control key creation
- Compiling and launching the URL

The URL must be generated (compiled) and launched by an external application. This is done by implementing support for URL-generation into the code of the external application.

5.1 Access control key creation

IDS7, LiteView and Order Management normally includes their own user access control functions, requiring a user to log on using a user name and password before access is granted. If however, the calling application includes its own reliable access control, the standard access control of the launched product can be bypassed. In that case the access control is made using a temporary access control key, supplied as a URL parameter.

5.1.1 Algorithm description

The temporary access control key is calculated on data identifying the requested examination combined with a timestamp and a system password (see section 3.2 About the system password).

The access control key is generated by computing the SHA1 hash of the concatenated URL parameter values, a time stamp and the system password.

The system password parameter (system_password) must be the last parameter in the concatenated string. The order of the other parameters is also of importance, as the access control key must be generated with the same order of the parameters as they are passed in the URL. If these orders differ, the access control key will be erroneous.

The following algorithm is used to calculate the access key:

```
string concatenatedParams = "";

concatenatedParams += stop;
concatenatedParams += user_id;
concatenatedParams += time;
concatenatedParams += pat_id;
concatenatedParams += acc_no;
concatenatedParams += exam_id;
concatenatedParams += his_id;
concatenatedParams += pat_birth;
concatenatedParams += pat_sex;
concatenatedParams += ref_unit;
concatenatedParams += mrn_integration_id;
concatenatedParams += acc_no_group;
concatenatedParams += sop_uid;
concatenatedParams += frame_number;
concatenatedParams += cmd;
concatenatedParams += ext_acc_no;
concatenatedParams += ids7_cmds;

// System password last
concatenatedParams += system_password;

byte[] utf8Bytes = GetUtf8Bytes(concatenatedParams);

byte[] sha1Hash = ComputeSHA1Hash(utf8Bytes);

string accessKey = ByteArrayToHexString(sha1Hash);
```

The input variables `pat_id`, `acc_no`, `ext_acc_no`, `exam_id`, `mrn_integration_id`, `acc_no_group`, `user_id`, `sp`, `pat_name`, `pat_birth`, `pat_sex`, `ref_unit`, `his_id`, `sop_uid` and `ids7_cmds` are strings corresponding to the parameter values of the URL after escape sequence substitution.

Note that escape sequence substitution should not be applied to the system password.

The implementation of “ `GetUtf8Bytes` ”, “ `ComputeSHA1Hash` ” and “ `ByteArrayToHex` ” will vary depending on programming language.

Note: If the `close_popup` parameter is used, it should not be included in the concatenated string.

5.1.2 Example execution

This section shows how the access control key creation algorithm executes, step-by-step, for the following example.

- user_id is “Dr.X”
- pat_id is “12121212-1212”
- Current time is 1285168669 (decimal string representation of the return value from the ANSI-C `time()` function, see section 3.4 About time limited URL)
- The system password is “test”

The following section shows the step-by-step results of the key calculations:

```
string parameters = "";

parameters += "Dr.X"; // "user_id", parameters is "Dr.X" after this operation
parameters += "1285168669"; // "time", parameters is "Dr.X1285168669"
                  // after this operation
parameters += "12121212-1212"; // "pat_id"
parameters += "test"; // system password, always last

// parameters are now "Dr.X128516866912121212-1212test"

byte[] utf8Bytes = GetUtf8Bytes(parameters);

// UTF-8 encode parameters. First byte will be 01000100 for 'D'

byte[] sha1Hash = ComputeSHA1Hash(utf8Bytes);

// Encrypt using SHA1. First two will be 01001110 and 11101010
```

```
 string accessKey = ByteArrayToHexString(sha1Hash);

 // Convert each byte to a hexadecimal string, e.g. 78 for 01111000 and
// 0d for 00001110, and compile a single lowercase string of them.
// accessKey is "4eea7f3d8ebde525197f45b7dcfae0e0cca325a3"
```

The resulting URL is

```
<protocol>://<hostname>/<webservice>?user_id=Dr.X&time=1285168669&
key=4eea7f3d8ebde525197f45b7dcfae0e0cca325a3&pat_id=121212-1212
```

5.2 Compiling and launching the URL

With the access control key generated, what's left is to compile and launch the URL. This section contains a full implementation of the EPR Integration (URL) flow, from access control generation to URL compilation and launching using both POST and GET requests, written in C#. For other examples of access control key generation, see appendix A Algorithm descriptions in an alternative programming language. Regarding launch of the URL, this is the only sample offered.

5.2.1 About the example implementation

The code contains two helper classes, `UrlLaunchString` and `WebLauncher`, and a simple program that sets up a `UrlLaunchString` and then launches it through a `WebLauncher`.

The best way to examine the code is to look at it in an IDE. Following is an example how to do it.

▶ **Building the code in Microsoft Visual Studio.**

1. Create a new project and choose `Console Application`.
2. Add project references to the following .NET assemlies: `System.Web`, `System.Drawing` and `System.Windows.Forms`.
3. Replace the code in `Program.cs` with the code in section 5.2.2 Example implementation and then build.
4. Add a break point in the beginning of the program's `Main() method` and start debugging by hitting F5.

5.2.2 Example implementation

```
using System;
using System.Text;
using System.Web;
using System.Collections.Generic;
using System.Security.Cryptography;
using System.Windows.Forms;

namespace UrlIntegration {

    class Program {

        [STAThread]
        static void Main(string[] args) {
```

```
        UrlLaunchString urlLaunchString = new UrlLaunchString() {
            UrlHost = "localhost",
            UrlPath = "OrderManagement",
            UseHttps = true,
            UseHttpPost = true,
            UserId = "Dr.X",
            PatientId = "12121212-1212",
            Stop = false,
            SystemPassword = "test"
        };

        WebLauncher launcher = new WebLauncher(urlLaunchString);
        launcher.Launch();
    }
}

internal class WebLauncher {

    public UrlLaunchString LaunchString { get; private set; }
    private Timer closeTimer = new Timer();
    private WebBrowser webBrowser;
    private Form webBrowserForm;

    public WebLauncher(UrlLaunchString launchString) {
        LaunchString = launchString;
    }

    public void Launch() {
        if (LaunchString.UseHttpPost) {
            LaunchHttpPost();
        } else {
            LaunchHttpGet();
        }
    }

    /// <summary>
    /// Launches the content of the URL launch string using http GET
    /// </summary>
    private void LaunchHttpGet() {
        string url =
            LaunchString.GetStartUrl() + "?" + LaunchString.GetQueryString();
        System.Diagnostics.Process.Start(url);
    }

    /// <summary>
    /// Launches the content of the URL launch string using http POST
    /// </summary>
    private void LaunchHttpPost() {
        // Create web browser control
        webBrowser = new WebBrowser();
        webBrowser.Visible = false;

        // Create form to host the web browser
        webBrowserForm = new Form();
        webBrowserForm.Size = new System.Drawing.Size(0, 0);
        webBrowserForm.FormBorderStyle = FormBorderStyle.None;
        webBrowserForm.Controls.Add(webBrowser);
        webBrowserForm.Visible = false;

        // Prepare post data
        byte[] postData =
            Encoding.UTF8.GetBytes(LaunchString.GetQueryString());
        string additionalHeaders =
            "Content-Type: application/x-www-form-urlencoded"
            + Environment.NewLine;

        // Set timer to close form in 3 seconds. That should be
        // enough to send the request to the default browser.
        closeTimer.Tick += closeTimer_Tick;
```

```
            closeTimer.Interval = 3000;
            closeTimer.Start();

            // Navigate to the url and post the data. The "_blank"
            // frame target will force opening the URL in the
            // standard browser of the workstation.
            webBrowser.Navigate(new Uri(LaunchString.GetStartUrl()),
                "_blank", postData, additionalHeaders);

            Application.Run(webBrowserForm);
        }

        /// <summary>
        /// When the close timer ticks, close all open forms.
        /// </summary>
        private void closeTimer_Tick(object sender, EventArgs e) {
            webBrowserForm.Close();
        }
    }

    internal class UrlLaunchString {

        /// <summary>
        /// These fields govern the URL format
        /// </summary>
        public bool UseHttps { get; set; }
        public bool UseHttpPost { get; set; }
        public string UrlHost { get; set; }
        public string UrlPath { get; set; }

        /// <summary>
        /// These fields govern the access control key
        /// </summary>
        public bool Stop { get; set; }
        public string UserId { get; set; }
        public string PatientId { get; set; }
        public List<string> AccessionNrs { get; set; }
        public List<string> ExamIds { get; set; }
        public string MrnIntegrationId { get; set; }
        public string AccessNumberGroup { get; set; }
        public string PatientName { get; set; }
        public DateTime? PatientBirthDate { get; set; }
        public string PatientSex { get; set; }
        public string ReferringUnit { get; set; }
        public string HisId { get; set; }
        public string SystemPassword { get; set; }
        public string Command { get; set; }
        public string ExternalRequestId { get; set; }
        public DateTime? DateAndTime { get; set; }
        public bool AllowPatientChange { get; set; }
        public List<string> IDS7Commands { get; set; }

        public UrlLaunchString() {
            AccessionNrs = new List<string>();
            ExamIds = new List<string>();
            IDS7Commands = new List<string>();
        }

        /// <summary>
        /// Compiles the URL fields into a start URL
        /// </summary>
        public string GetStartUrl() {
            StringBuilder url = new StringBuilder();
            string protocol = this.UseHttps ? "https://" : "http://";
            url.Append(protocol);
            url.Append(this.UrlHost + "/");
            url.Append(this.UrlPath + "/3pstart.aspx");

            return url.ToString();
        }
        /// <summary>
        /// Adds a string to the parameters to be added
        /// </summary>
        public void AddStringParam(string input, string keyName,
            ref List<string> hashParams, ref List<string> retval) {
            if (String.IsNullOrEmpty(input)) return;
            var escapeStr = EscapeString(input);
            hashParams.Add(escapeStr);
            retval.Add(keyName + "=" + escapeStr);
        }

        /// <summary>
        /// Adds a list to the parameters to be added
        /// </summary>
        public void AddListParam(List<string> input, string keyName,
            ref List<string> hashParams, ref List<string> retval) {
            if (input == null || input.Count <= 0) return;
            var parameterString = ListToString(input, '^', true);
            AddStringParam(parameterString, keyName, ref hashParams, ref retval);
        }

        /// <summary>
        /// Compiles the UrlLaunchString into a query string
        /// </summary>
        public string GetQueryString() {
            List<string> retval = new List<string>();
            List<string> hashParams = new List<string>();

            if (Stop) {
                retval.Add("stop=1");
                hashParams.Add("1");
            }
            AddStringParam(UserId, "user_id", ref hashParams, ref retval);
            AddStringParam(PatientId, "pat_id", ref hashParams, ref retval);
            AddListParam(AccessionNrs, "acc_no", ref hashParams, ref retval);
            AddListParam(ExamIds, "exam_id", ref hashParams, ref retval);
            AddStringParam(MrnIntegrationId, "mrn_integration_id", ref hashParams, ref retval);
            AddStringParam(AccessNumberGroup, "acc_no_group", ref hashParams,
                ref retval);
            AddStringParam(PatientName, "pat_name", ref hashParams, ref retval);
            if (PatientBirthDate != null) {
                var birthDate = PatientBirthDate.Value.ToString("yyyyMMdd");
                AddStringParam(birthDate, "pat_birth", ref hashParams, ref retval);
            }
            AddStringParam(PatientSex, "pat_sex", ref hashParams, ref retval);
            AddStringParam(ReferringUnit, "ref_unit", ref hashParams, ref retval);
            AddStringParam(HisId, "his_id", ref hashParams, ref retval);
            AddStringParam(Command, "cmd", ref hashParams, ref retval);
            AddStringParam(ExternalRequestId, "ext_acc_no", ref hashParams,
                ref retval);

            if (AllowPatientChange) {
                retval.Add("allow_pat_change=1");
                hashParams.Add("1");
            } else {
                retval.Add("allow_pat_change=0");
                hashParams.Add("0");
            }

            AddListParam(IDS7Commands, "ids7_cmds", ref hashParams, ref retval);

            if (!String.IsNullOrEmpty(SystemPassword) &&
                !String.IsNullOrEmpty(UserId)) {
                TimeSpan t = (DateTime.UtcNow - new DateTime(1970, 1, 1));
                long time = (long)t.TotalSeconds;
                var timeString = time.ToString();
                AddStringParam(timeString, "time", ref hashParams, ref retval);

                // System password must be added last to the parameters
                // since it will be added last by the part verifying the
                // launch string.
                hashParams.Add(SystemPassword);
                retval.Add("key=" + GetAccessControlKey(hashParams));
            }

            return ListToString(retval, '&', false);
        }

        /// <summary>
        /// Strings might contain characters that are not supported in
        /// a URL, so these need escaping.
        /// </summary>
        private string EscapeString(string str) {
            return HttpUtility.UrlEncode(str);
        }

        /// <summary>
        /// Compile a string with list items separated by a separator
        /// character.
        /// </summary>
        private string ListToString(List<string> strList,
            char separator, bool escape) {

            StringBuilder retval = new StringBuilder();
            foreach (string str in strList) {
                if (retval.Length > 0) {
                    retval.Append(separator);
                }
                if (escape) {
                    retval.Append(EscapeString(str));
                } else {
                    retval.Append(str);
                }
            }

            return retval.ToString();
        }

        /// <summary>
        /// Generate the access control key from a list of parameters.
        /// </summary>
        private string GetAccessControlKey(IEnumerable<string> hashParameters) {
            if (hashParameters == null) {
                return String.Empty;
            }
            StringBuilder concatenatedParams = new StringBuilder();
            foreach (string param in hashParameters) {
                concatenatedParams.Append(param);
            }

            // The three steps from string of params to hexadecimal
            // string of SHA1 hash bytes.
            byte[] utf8Bytes = GetUtf8Bytes(concatenatedParams.ToString());
            byte[] sha1Hash = ComputeSHA1Hash(utf8Bytes);
            string accessControlKey = ByteArrayToHexString(sha1Hash);

            return accessControlKey;
        }

        /// <summary>
        /// Returns a byte array representing each character in
        /// the input string as a UTF-8 encoded byte.
        /// </summary>
        private static byte[] GetUtf8Bytes(string parameters) {
            return Encoding.UTF8.GetBytes(parameters);
        }

        /// <summary>
        /// Computes a hashed byte array of the input using the
```

```
        /// SHA-1 hash algorithm.
        /// </summary>
        private static byte[] ComputeSHA1Hash(byte[] utf8bytes) {
            SHA1Managed hasher = new SHA1Managed();
            return hasher.ComputeHash(utf8bytes);
        }

        /// <summary>
        /// Converts a byte array into a hexadecimal lowercase
        /// string of the concatenated bytes.
        /// </summary>
        private static string ByteArrayToHexString(byte[] sha1Hash) {
            StringBuilder hexString = new StringBuilder();
            foreach (byte hashByte in sha1Hash) {
                hexString.Append(hashByte.ToString("x2"));
            }

            return hexString.ToString();
        }
    }
}
```

6 About External Thumbnail Viewer

The following topics are included in this chapter:

A client application using the external thumbnail web page will do so by calling a URL pointing to an ASP.NET web page hosted by a Sectra Healthcare Server or a Sectra Satellite Server. This URL is built from two parts:

1. The first part points out the Sectra Healthcare Server or Sectra Satellite Server and External Thumbnail Viewer web page.
2. The URL integration parameters are appended with a sequence of parameters used to control thumbnail presentation specific characteristics such as colors.

The result from the external thumbnail web page call is an HTML stream that the client is responsible for displaying. A web application client could for example host the thumbnail web page in a frame set frame, while a WinForm application could host the thumbnail web page in a WebControl.

The `ViewerExternalThumbnails.exe` application can be used to see what the ASP.NET web page looks like within an application and can also be used for testing different parameter settings. See section 6.4 About ViewerExternalThumbnails.exe.

Note: The chapter about closing the viewer application is essential to determine how to integrate the External Thumbnail Viewer.

6.1 About the External Thumbnail Viewer display

The result from a thumbnail web page call could be one of these alternatives:

- **Empty** - No thumbnails were found. The resulting web page will simply alert the user of this.
- **Displaying thumbnails for an examination** - There will be one header for each WISE server returning thumbnails for the requested procedure. Following each header, the thumbnail images are presented. If only one WISE server returns thumbnails, no header is shown.
- **Empty, but with a link that allows launch of the external viewer** - This is the case if more than one examination has been specified for display.

The mode is automatically selected, depending on what the control is asked to display.

6.2 About starting of IDS7 or LiteView

When a user clicks on the thumbnail images either IDS7 or LiteView is launched to show the full image. What application, IDS7 or LiteView, should be launched when clicking a thumbnail is controlled by provided parameters, but may also be controlled using advanced configuration, see System Administrator's Guide Sectra Healthcare System [3]

Note: Values sent in integration parameters to control the behavior of the thumbnails will have precedence over values set in advanced configuration.

Note: For IDS7 to launch without any extra dialogs, and in some cases for the launch to work at all, the following configuration is sometimes necessary on the workstations: In the **Internet Options** dialog of Internet Explorer, select the **Security** tab and check the settings for the zone that the Sectra Healthcare Server belongs to. The setting **Downloads**->**Automatic prompting for file downloads** needs to be **Enabled**.

6.3 About closing the viewer application

The behaviour when closing the External Thumbnail Viewer depends on the launched application.

LiteView External Thumbnail Viewer will automatically close LiteView.

IDS7 External Thumbnail Viewer can not close IDS7 automatically, this must be done by the client application.

When IDS7 is launched, the client application can send stop commands in two ways.

▶ To always send stop command

1. The client application needs to specify the launch url when both starting and closing the viewer application see `tc_launch_url` in Table 7.2, “Parameters for control of the thumbnail web page behavior and look.” for more information.

▶ To send stop command only if IDS7 has been launched

Note: The client application must host the External Thumbnail Viewer in a web page belonging to the same domain.

1. The global variable `isIDS7Launched` should be checked to determine if IDS7 has been launched.
2. The global variable `stopUrl` should be used to determine the address to use to close the viewer application.
3. Send the stop command to the url retrieved in the previous step, this is best done when the `onbeforeunload` event is triggered.

Note: In releases prior to 15.1 this was handled by the client application using the EPR Integration (URL).

6.4 About ViewerExternalThumbnails.exe

The application `ViewerExternalThumbnails.exe` makes it possible to test both the External Thumbnail Viewer functionality and launching of Order Management using EPR Integration (URL). The `ViewerExternalThumbnails.exe` application is a pre-compiled application that is included in the SDK.

To be able to run the ViewerExternalThumbnails.exe application all files in the bin folder (SDK) must be copied to the local disk of the computer.

Important: Please note that the `ViewerExternalThumbnails.exe` application is provided as is and is only to be used for test purposes.

ViewerExternalThumbnails.exe is a graphical interface application. The list below contains the parameters that can be configured in the `Administration` and `Get thumbnails` tabs of `ViewerExternalThumbnails.exe`:

Server The host name or ip address of the Sectra Healthcare Server or Sectra Satellite Server.

Web application Either Sectra Healthcare Server or Sectra Satellite Server.

SSL If Secure Sockets Layer is used or not for the link to launch the viewer application.

URL integration system password The system password for URL configured in Sectra Enterprise Manager. Used both for External Thumbnail Viewer and EPR Integration (URL) to Order Management.

Patient id A patient id of a known patient that exists in the Sectra PACS and has at least one examination with images. Log on to IDS7 to find an example.

User id A user name id of a known user in the Sectra PACS or a new user that will be created by automatic creation of user accounts.

Accession number An accession number of an existing examination for the patient. Multiple accession numbers are separated by a ‘^’ character. The accession and examinations must be given in pairs. If several examinations with the same accession number exist, the same accession number should be entered several times separated by a ‘^’.

Note: If multiple examinations are asked for, the response will not be any thumbnail images but a text message saying: 'More than one examination was found' and a link to be able to launch IDS7 or LiteView.

Exam id An examination id of the existing examination. Multiple examination numbers are separated by a ‘^’ character. The accession and examinations must be given in pairs. If several examinations with the same accession number exist, the same accession number should be entered several times separated by a ‘^’.

Note: If multiple examinations are asked for, the response will not be any thumbnail images but a text message saying: 'More than one examination was found' and a link to be able to launch IDS7 or LiteView.

MRN Integration Id
If medical record number integration ids are configured in the Sectra Enterprise Manager, enter the one used by the system where the patient exists.

Accession number group
If accession number group(s) are configured in Sectra Enterprise Manager, enter the one used by the system where the examination exists.

IDS7 command(s)
A list of commands separated by ‘^’ that should be run when IDS7 is started.

Locale
Name of language on the .NET culture info class format. Examples: en-US or sv-SE.

Debug mode
If debug mode is enabled or not. If enabled some debug information will be written in the window beside the thumbnail images.

Launch on click
If the viewer application should be launched or not when clicking the thumbnail images.

Launch URL
A custom URL to launch when clicking a thumbnail.

Launch behavior
Determines how a clicked thumbnail should open in the browser.

Header foreground color
Hex color value of foreground color without any '#' character.

Header background color
Hex color value of background color without any '#' character.

Header border color
Hex color value of border without any '#' character.

Page background color
Hex color value of background without any '#' character.

Font family
Name of the fonts to be used.

Host
The Sectra Workflow Server to be used when using ViewerExternalThumbnails.exe to test launch Order Management.

OM user
The user to be used when using ViewerExternalThumbnails.exe to test launch Order Management. See section 4.1 URL parameter descriptions for formatting.

The predefined settings in ViewerExternalThumbnails.exe are stored as a XML file in the folder where the application is started. The name of the XML file is `user.config`.

6.5 About starting of Order Management

When using ViewerExternalThumbnails.exe for the purpose of testing Order Management, make use of the `Read reports` tab. The host, OM user and URL integration system password are configured in the `Administration` tab, see section 6.4 About ViewerExternalThumbnails.exe, but in order to launch Order Management, make sure to use the `Create request for patient` button on the `Read reports` tab instead of the `Step 1. Generate URL` and `Step 2. Get thumbnails` buttons, available on the `Get thumbnails` tab.

6.6 Multiple server configuration

If cross-platform workflow is enabled and there exist several WISE servers within the system, thumbnails can be viewed for examinations available both on the local system as well as remote system. This can be done without any further configuration of External Thumbnail Viewer.

However there are three settings controlling the look of the header separating thumbnails from different WISE servers: **tc_headfcol**, **tc_headbcol** and **tc_backgroundcol**. See Table 7.2, “Parameters for control of the thumbnail web page behavior and look.” for more information.

Note: To show thumbnails from remote systems, the role that the automatically created users will belong to must be configured to search multiple servers. To do this, open the IDS7 options dialog for that role, select the **Worklist** tab and click **Server Options...**.

6.7 About backward compatibility

The External Thumbnail Viewer was earlier known as the Thumbnail Control.

For IDS5/web there existed an add-in to the URL integration called the thumbnail control. It was an ActiveX component, `thumbnail.ocx`, that could display the thumbnail images for an examination. It was typically integrated into the external application. When the user clicked on a thumbnail image, the URL integration was used to launch IDS5/web displaying the selected image. The External Thumbnail Viewer is remodeled in versions 12.2 and therefor any old integration needs to be modified.

Note: The External Thumbnail Viewer is remodeled in versions 12.2.

Note: The environment variable `IDS7_URLINTEGRATION_SITE` has been deprecated in version 15.1. In previous versions this variable could be set to override the web site name from where IDS7 would be launched. Integrations for 15.1 and later should use integration parameter **tc_launch_url** or advanced configuration.

7 About the URL for External Thumbnail Viewer

The following topics are included in this chapter:

- Parameters description
- URL parameter encoding
- URL examples

The URL is a compiled string that specifies the patient and examination of the thumbnails. It also specifies the look of the web page and how clicked thumbnails should be launched.

Note: The External Thumbnail Viewer only applies to IDS7 or LiteView and not to Order Management.

These are the main parts of the URL:

```
<protocol>://<hostname>/<webservice>?<parameters>
```

<protocol> http or https depending on the configuration of the web-server.

<hostname> The hostname of the Sectra Healthcare Server or the Sectra Satellite Server from where the thumbnails will be fetched.

<webservice> The path to the web service of Sectra Healthcare Server or Sectra Satellite Server. The default webservice is *Instance/SectraHealthcareServer/Web/Thumbnails/GetThumbnails.aspx* where instance is the name of the Sectra Healthcare Server instance. GetThumbnails.aspx is the very web page executed by IIS that will serve thumbnail images to clients calling.

<parameters> Several parameters can be passed on to specify the patient and examination of the thumbnails. It also specifies the look of the web page and how IDS7 or LiteView should be launched.

7.1 Parameters description

In Table 7.1, “URL integrations parameter description” the patient, examination and other configuration parameters are listed. The parameters supplied must be encoded as specified in section 7.2 URL parameter encoding.

Table 7.1 URL integrations parameter description

Key name	Sample value	Description	Mandatory
`pat_id`	`193201011234`	Id of current patient	Yes
`user_id`	`EL-PRE-79`	Either a user known to Sectra PACS, or a user only known by the client application. In the latter case use Sectra Enterprise Manager to configure the WISE server to 'Automatically create users'. See also section 3.3 About user handling	Yes
`acc_no`	`HL70000000000001`	Accession number. Multiple accession numbers are separated by a '^' character. The accession and examinations must be given in pairs. If several examinations with the same accession number exist, the same accession number should be entered several times separated by a '^'. If multiple accession numbers are asked for, the response will not be any thumbnail images but a text message saying: 'More than one examination was found' and a link to be able to launch IDS7 or LiteView.	Yes
`exam_id`	`HL70000000000001`	Examination id. Multiple exam id are separated by a '^' character. The accession and examinations must be given in pairs. If several examinations with the same accession number exist, the same accession number should be entered several times separated by a '^'. If multiple examinations are asked for, the response will not be any thumbnail images but a text message saying: 'More than one examination was found' and a link to be able to launch IDS7 or LiteView.	Yes
`mrn_integration_id`	`Default`	Medical Record Number (MRN) integration ID. This value is combined with the patient ID to provide a unique patient identifier. It is defined for each Medical Number Group in Sectra Enterprise Manager, under the **Servers** node for the Sectra Healthcare Server.	Yes
`acc_no_group`	`local_acc_no_group`	Accession number group. This property is defined in Sectra Enterprise Manager, on the properties page, general tab, for the WISE server.	Yes
`allow_pat_change`	`0`	The flag will be forwarded to the launched viewer application. For description of the flag, see section 4.1 URL parameter descriptions.	No
`time`	`1250087673`	Time of call expressed as number of milliseconds since January 1st 1970. The time stamp should not be compensated for daylight saving.	Yes
`key`	`C009E14`	Hash key computed from all parameters but the ones starting with 'tc_', i.e. all parameters in this table. The computation algorithm is described in section 5.1 Access control key creation.	Yes

There are also some optional parameters controlling the look of the web page. In Table 7.2, "Parameters for control of the thumbnail web page behavior and look." these parameters are listed.

Table 7.2 Parameters for control of the thumbnail web page behavior and look.

Key name	Sample value	Description	Default value	Mandatory
`tc_lng`	`sv-SE`	Language of the thumbnail page. This is the same format that is used in the .NET CultureInfo class. The parameter will be used if it exists, otherwise the 'UserLanguages' field in the http request header is used and if also this is missing the default value en-US will be used.	en-US	No
`tc_ssl`	`0`	Indicates if the launch should be handled using secure socket layer or not. 0 – no ssl, 1 - use ssl	0	No
`tc_launch`	`1`	Indicates whether the user can open a full size image by clicking the corresponding thumbnail image. 0 – no launch, 1 - launch	0	No
`tc_launch_url`	`http://servername/IDS7/3pstart.aspx`	A custom url to launch when clicking a thumbnail. All other parameters will be appended to the url. Example url for launching IDS7 is http://servername/IDS7/3pstart.aspx. Example url for launching LiteView is https://servername/LiteView/index.html#launch. **Note:** *A value for this parameter overrides the value set in advanced configuration in Sectra Enterprise Manager.* **Note:** *A value for this parameter requires a value for `tc_launch_behavior`.*	http://thumbnailserver/IDS7/3pstart.aspx where 'thumbnailserver' is the server serving the thumbnails	No
`tc_launch_behavior`	`ISD7`	Determines how a clicked thumbnail should open in the browser. 'IDS7' is suitable when launching IDS7/cx to prevent a browser window to open. 'LiteView' is recommended for LiteView to let thumbnails open in the same new browser window. **Note:** *A value for this parameter overrides the value set in advanced configuration in Sectra Enterprise Manager.* **Note:** *A value for this parameter requires a value for `tc_launch_url`.*	IDS7	No
`tc_headfcol`	`216194`	Foreground color of WISE server header Expressed as hex color value, omitting the '#' sign. See section 6.6 Multiple server configuration.	White	No
`tc_headbcol`	`CEDBDE`	Background color of WISE server header Expressed as hex color value, omitting the '#' sign. See section 6.6 Multiple server configuration.	Gray	No
`tc_headbordercol`	`206090`	Border color of WISE server header Expressed as hex color value, omitting the '#' sign. See section 6.6 Multiple server configuration.	Black	No
`tc_backgroundcol`	`D5D5D5`	Background color of the entire thumbnail web page. Expressed as hex color value, omitting the '#' sign.	White	No

Key name	Sample value	Description	Default value	Mandatory
`tc_font`	`Verdana, Arial, Helvetica`	Fonts in priority order. If the first one (Verdana) is not installed on the client, then the second one (Arial) is selected and if that fails as well the third one (Helvetica) is chosen. If no font is installed on the client machine, the HTML default font is selected as defined by the surrounding client web application or the browser.	HTML default font	No
`tc_disabledatauri`	`0`	IE8 (and later) has support for a protocol called data uri. This means images are included in the first reply and only one roundtrip to the server is required. The default value is '0', which means the server will identify the browser used and return html using data uri if possible. If tc_disabledatauri = '1', data uri will never be used and two roundtrips to the server are required.	0	No

7.2 URL parameter encoding

The parameter values of the URL are only allowed to contain characters listed in Table 4.2, "Allowed characters in parameter values". The list is the same as for EPR Integration (URL). All other characters must be substituted with an escape sequence.

7.3 URL examples

To view thumbnail images from one examination:

```
http://myserver/Instance/SectraHealthcareServer/Web/Thumbnails/GetThumbnails.aspx?
pat_id=193201011234&user_id=EL-PRE-79&tc_lng=sv-SE&tc_ssl=0&tc_dbg=1&
acc_no=HL70000000000001&exam_id=HL70000000000001&mrn_integration_id=Default&
acc_no_group=local_acc_group&time=1250087673&tc_launch=1&
tc_headfcol=216194&tc_headbcol=CEDBDE&tc_headbordercol=206090&
tc_backgroundcol=D5D5D5&key=C009E14&
```

To get a link to launch the viewer application and a text saying 'More than one examination was found' when asking for two different examinations with the same accession number:

```
http://localhost/SectraHealthcareServer/Web/Thumbnails/GetThumbnails.aspx?
pat_id=191212121212&user_id=jo-sjo&acc_no=InstA0000001^InstA0000002&
exam_id=InstA0000001^InstA0000002&mrn_integration_id=Default&
acc_no_group=localhost_dsid_debug&
time=1254747313&key=0a33d8da0650d71e70a73288ddfc6cb1745efca6&tc_lng=SE-Sv&
tc_ssl=1&tc_dbg=1&tc_launch=1&tc_headfcol=D5D5D5&tc_headbcol=AEDBD5&
tc_headbordercol=D5D5D5&tc_backgroundcol=AEDBD5&tc_font=Ariel
```

Use the generate URL function in `ViewerExternalThumbnails.exe` to get some more examples of valid URL strings. For information about ViewerExternalThumbnails.exe see section 2.8 Launching IDS7 or LiteView from the External Thumbnail Viewer for the first time.

8 How to compile the URL for External Thumbnail Viewer

The URL must be generated (compiled) and launched by an external application. See chapter 7 About the URL for External Thumbnail Viewer for information about the URL.

A Algorithm descriptions in an alternative programming language

The following topics are included in this appendix:

- Java
- Javascript for External thumbnails

This appendix contains program listings of the algorithms mentioned in this document in an alternative programming language, e.g. Java. For an implementation in C#, see section 5.2.2 Example implementation.

Important: The program listings in this appendix are provided only as examples. Sectra AB takes no responsibility whatsoever regarding the actual use in external applications of the code presented herein.

A.1 Java

The following program listing gives an example on how to implement the algorithms mentioned in this document using Java:

Note: This is an incomplete code sample, showing only how to create the key.

```
import java.net.URLEncoder;
import java.security.MessageDigest;
import java.util.Formatter;

public class GenerateKey {

  public static void main(String[] args)
    throws java.io.UnsupportedEncodingException,
    java.security.NoSuchAlgorithmException {

    StringBuilder parameters = new StringBuilder();
    parameters.append(URLEncoder.encode("Dr.X", "UTF-8"));
    parameters.append(URLEncoder.encode("12121212-1212", "UTF-8"));
    parameters.append(URLEncoder.encode("1234", "UTF-8"));
    parameters.append("^");
    parameters.append(URLEncoder.encode("2345", "UTF-8"));
    parameters.append(URLEncoder.encode("001", "UTF-8"));
    parameters.append("^");
    parameters.append(URLEncoder.encode("001", "UTF-8"));

    // You need to handle getting the proper amount of seconds
    // since 1970-01-01 below.
    parameters.append(URLEncoder.encode("1065507963", "UTF-8"));
    parameters.append("test");

    // The URLEncoder will create uppercase hex, but
    // PACS expects lowercase (which is default
    // for url encoding in Microsft.Net).
    // You need to handle this in an implementation of
    // the method below in your code.
    ConvertUrlEncodedHexToLowerCase(parameters);

    MessageDigest sha1 = MessageDigest.getInstance("SHA1");
    byte[] hash = sha1.digest(parameters.toString().getBytes("UTF-8"));

    Formatter formatter = new Formatter();
    for (byte hashByte : hash) {
      formatter.format("%02x", hashByte);
    }

    System.out.println(formatter.toString());
  }
}
```

A.2 Javascript for External thumbnails

The following program listing gives an example on how to retrieve and display thumbnail images on a web page, using Javascript.

Note: This is an incomplete code sample. The SHA1 encryption function must be downloaded from www.webtoolkit.info (search for “sha1”).

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">

<head>
    <title>Sectra External Thumbnails Demonstrator</title>
    <style type="text/css">
        span { display: inline-block; width: 200px; }
        input { width: 300px; }
        .button { width: 100px; }
    </style>
</head>

<body>
    <div>
        <h3>Sectra External Thumbnails Demonstrator</h3>
        <div>
            <div><span>Server url:</span><input id="serverurl" type="text"
               value="http://myserver"  /></div>
            <div><span>Accession number group:</span><input id="accnogroup" type="text"
               value="myaccnogroup"  /></div>
            <div><span>Medical record number group:</span><input id="mrngroup" type="text"
               value="mymrngroup"/></div>
            <div><span>User:</span><input id="user" type="text"
               value="myuser" /></div>
            <div><span>Allow patient change:</span><input id="patchange" type="text"
               value="0" /></div>
            <div><span>SSL:</span><input id="ssl" type="text"
               value="0" /></div>
            <div><span>Allow launch:</span><input id="launch" type="text"
               value="1"  /></div>
            <div><span>System password:</span><input id="password" type="password"
               value="mysystempassword"  /></div>
            <div><span>Medical record number:</span><input id="mrn" type="text"
               value="patientmrn" /></div>
            <div><span>Accession number:</span><input id="accno" type="text"
               value="accessionnumber" /></div>
            <div><span>Examination number:</span><input id="examno" type="text"
               value="examinationnumber" /></div>

            <input class="button" type="button" id="btnLoad" value="Load" onclick="loadThumbs()" />
            <hr />

            <div><span class="label" >Generated url:</span><textarea id="url" style="width:100%;
               background-color: lightgray" readonly rows="4">-</textarea></div>
        </div>

        <div>
            <!-- Always put the thumbnails in an iframe, as the thumbnails are returned from SHS
            as an entire html document. -->
            <iframe id="thumbnailContainer" width="100%" allowTransparency="true" frameborder="0"
               scrolling="no" src="" type="height: 1000px">Not supported.</iframe>
        </div>

    </div>

    <script type="text/javascript">

        function valueFor(elementId) {
            return document.getElementById(elementId).value;
        }

        function getTimestamp() {
            var t = new Date().getTime(); //milliseconds since 1970-01-01 UTC
            var time = Math.floor(t / 1000); //  seconds since 1970-01-01 UTC
            return time;
        }

        function calcKey(timestamp, encodedUser) {
            var hashString =
                valueFor("mrn") +                   // 1st item in url
                valueFor("accno") +                 // 2nd item in url
                valueFor("examno") +                // 3rd item in url
                valueFor("mrngroup") +              // 4th item in url
                valueFor("accnogroup") +            // 5th item in url
                encodedUser +                       // 6th item in url
                valueFor("patchange") +             // 7th item in url
                timestamp +                         // 8th item in url
                valueFor("password");               // Always final item , not part of the url.

            return SHA1(hashString);
        }

        function loadThumbs() {

            var encodedUser = encodeURIComponent(valueFor("user")); // uri encode when required.
            var timestamp = getTimestamp();
            var url = valueFor("serverurl") +

                "/SectraHealthcareServer/Web/Thumbnails/GetThumbnails.aspx?" +

                // Non TC parameters can be placed in any order.
                "pat_id=" + valueFor("mrn") + "&" +                 // 1st non TC parameter
                "acc_no=" + valueFor("accno") + "&" +               // 2nd non TC parameter
                "exam_id=" + valueFor("examno") + "&" +             // 3rd non TC parameter
                "mrn_group=" + valueFor("mrngroup") + "&" +         // 4th non TC parameter
                "acc_no_group=" + valueFor("accnogroup") + "&" +    // 5th non TC parameter
                "user_id=" + encodedUser + "&" +                    // 6th non TC parameter
                "allow_pat_change=" + valueFor("patchange") + "&" + // 7th non TC parameter
                "time=" + timestamp + "&" +                         // 8th non TC parameter
                // Additional non tc_ parameters goes here

                // After all non TC parameters comes the key
                "key=" + calcKey(timestamp, encodedUser) + "&" +

                // TC parameters not part of the hash, place last in any order
                "tc_ssl=" + valueFor("ssl") + "&" +
                "tc_launch=" + valueFor("launch");
                // Additional tc_ parameters goes here...

            // Display the generated url.
```

```
            document.getElementById("url").value = url;

            // Set the iframe url, this will render a navigate event and the thumbnails are retreived.
            document.getElementById("thumbnailContainer").setAttribute("src", url);
        }

        /**
        *
        *  TODO:
        *  Paste Secure Hash Algorithm (SHA1) here.
        *  Download from http://www.webtoolkit.info/
        *
        **/

    </script>

</body>

</html>
```

B EPR Integration (URL) to an MSI installed IDS7

The following topics are included in this appendix:

- The custom URL protocol

This appendix describes how EPR Integration (URL) works when having IDS7 installed as a local application from an MSI, instead of as a web-based application.

See System Administrator's Guide Sectra Healthcare System [3] for general information about MSI deployment.

B.1 The custom URL protocol

The chapters above describe EPR Integration (URL) using a URL starting with

```
http://<hostname>/IDS7/3pstart.aspx?<url-parameters>
```

This leads to a web service that launches a ClickOnce deployed IDS7 instance, and passes all parameters to it. When you want IDS7 to run from an installed location on the local workstation, this is not without complication since a web-page can not launch a local application on a Windows system. There are several reasons for this, the most obvious one being security.

To enable a link in e.g. a HIS to launch a local IDS7, a custom URL protocol is registered as part of the MSI installation. This means that any URL beginning with `ids7`: will be handled by IDS7, just like e.g. those beginning with `http:` are handled by the default web browser. A URL that should launch a locally installed IDS7 thus has the following form.

```
ids7:<url-parameters>
```

Note: Always implement integration using the `http:` URL format. As described below, the system will automatically redirect to an `ids7:` URL if there is a local IDS7 installed.

In order to not complicate things for the development of integrating applications, the `3pstart.aspx` web page transparently converts URLs from `http:` to `ids7:` if there is a local IDS7 installed on the workstation. This is detected using a custom entry in the user agent string of Internet Explorer, which is added during the MSI installation of IDS7. If the user agent contains the string `"Local IDS7"`, the custom URL protocol will be used.

Note: The MSI installation adds information about `ids7:` URLs to the registry that enables integration with a local IDS7. However, a reboot of the workstation is necessary for Internet Explorer to read and use this. Correspondingly, after IDS7 has been uninstalled, a reboot is necessary to make the system go back to always using ClickOnce for EPR Integration (URL).

Note: The MSI installation will store the URL to the Sectra Healthcare Server or Sectra Satellite Server in the registry under `HKLM\SOFTWARE\Sectra\Workstation\SHS_SERVER_URL`. If the server name has to be changed for some reason, it is possible to manually modify the registry value to point to the correct URL. This means that when launching the local application this will always be used and not the hostname in the URL given.

Tip: To investigate problems where a ClickOnce IDS7 is started instead of the locally installed one when using EPR Integration (URL), check that the registry key `HKLM\Software\Microsoft\Windows\CurrentVersion\Internet Settings\5.0\User Agent\Post Platform` contains a value called `Local IDS7`. For 64-bit systems, also check for the same value in `HKLM\Software\Wow6432Node\Microsoft\Windows\CurrentVersion\Internet Settings\5.0\User Agent\Post Platform`.

The quality system of Sectra AB [Sectra] conforms to ISO 9001, ISO 13485 and ISO 27001. All Sectra medical devices have obtained regulatory clearance for those markets where Sectra sells and deploys its devices, e.g. EEA, USA, Canada, Australia. For further regulatory information please contact Sectra.

Sectra, the Sectra logotype, the Sectra Imtec Handshake picture and the Sectra circles are registered trademarks of Sectra AB. Pronosco and Pronosco X-posure System are registered trademarks of Sectra AB. OrthoStation is a registered trademark of Sectra North America, Inc. Sectra PACS, Sectra Breast Imaging PACS, Sectra RIS, Sectra Screening RIS, Sectra OneScreen, IDS5, IDS7, WISE, ImageServer, Clinical Solutions Network, Sectra Medical Interfaces, Clinical Application Interface and Cross Platform Worklist are trademarks of Sectra AB. PACS-Guard is a servicemark of Sectra North America, Inc.

Windows is a registered trademark of Microsoft Corporation in the United States and other countries. *Mac*, *Safari* and *iPad* are trademarks of Apple Inc., registered in the U.S. and other countries. All other names/products by ® or ™ are registered trademarks of the respective manufacturer.

This product may be covered by U.S. patents 6,005,917; 6,226,393; 6,411,729; 6,763,257; 7,162,623; 7,532,214; 7,660,461; 7,689,539; 7,830,381; 7,936,930; 7,940,270; 7,992,100; 8,041,129; 8,051,386; 8,131,033; 8,295,620; Japanese patent 3646122; European patents 0818971; 1046374; and Swedish patents 0203545; 0300951. Patents Pending in the U.S. and other countries.

World Headquarters,
manufacturer
Sectra AB
Teknikringen 20
SE-58330 Linköping
Sweden
Phone: +46 13 23 52 00
E-mail: info@sectra.se

CE 0434

USA
Phone: +1 203 925 0899
E-mail: info.na@sectra.com

Germany/Switzerland/Austria
Phone: +49 221 47 457 0
E-mail: info.de@sectra.com

United Kingdom/Ireland
Phone: +44 1908 673 107
E-mail: info.uk@sectra.com

Japan
E-mail: info.jp@sectra.com

Sweden
Phone: + 46 13 23 52 00
E-mail: info.se@sectra.se

Norway
Phone: +47 67 58 97 70
E-mail: info.no@sectra.com

Denmark
Phone: +45 45 65 06 00
E-mail: info.dk@sectra.com

Spain/Portugal
Phone: +351 22 011 00 20
E-mail: info.iberia@sectra.com

Benelux
Phone: +31 36 540 1970
E-mail: info.benelux@sectra.com

Australia/New Zealand
Phone: + 61 2 9420 1620
E-mail: info.anz@sectra.com

SECTRA

For other regions please visit www.sectra.com/medical