SDK - ndd EMR plug-in

**Software Developer Guide**

**Warning:** Please note, that the developer assumes full liability for using their plug-in. The developer must verify functionality, in order to ensure patient safety and proper operation and must act in compliance with the appropriate medical regulations and standards.

**Document History**

| No | Date | Version | Modifications(s) | State | Editor(s) |
| --- | --- | --- | --- | --- | --- |
| 1 | 12.03.2014 | 01 | Initial version | Released | MZ |
| 2 | 21.05.2014 | 02 | Updated to EWP V.1.8.0.14 | Released | MZ |

Content

[1 Scope 3](#_Toc386117906)

[2 Conventions 3](#_Toc386117907)

[2.1 Abbreviations 3](#_Toc386117908)

[2.1.1 EMR , EHR and EPR 3](#_Toc386117909)

[2.1.2 EWP 3](#_Toc386117910)

[2.2 Format 3](#_Toc386117911)

[3 Introduction 4](#_Toc386117912)

[4 Design 4](#_Toc386117913)

[5 Requirements/Limitations 4](#_Toc386117914)

[6 How to get started 5](#_Toc386117915)

[7 Control Easy on-PC with your own XML file 7](#_Toc386117916)

[8 Create your own Plug-In 8](#_Toc386117917)

[9 XML structure 10](#_Toc386117918)

[10 Interface 11](#_Toc386117919)

[10.1 Plug-In interface 11](#_Toc386117920)

[10.2 Configuration 11](#_Toc386117921)

[11 Commands 12](#_Toc386117922)

[11.1 GetSupportedFeatures 12](#_Toc386117923)

[11.2 SupportedFeatures 12](#_Toc386117924)

[11.3 GetConfiguration 13](#_Toc386117925)

[11.4 Configuration 13](#_Toc386117926)

[11.5 SyncPatient 15](#_Toc386117927)

[11.6 AddToWorklist 16](#_Toc386117928)

[11.7 RemoveWorklist 17](#_Toc386117929)

[11.8 PerformTest 18](#_Toc386117930)

[11.9 TestResults 18](#_Toc386117931)

[11.10 GetTestResults 19](#_Toc386117932)

[11.11 ReviewTest 19](#_Toc386117933)

[11.12 PrintTest 20](#_Toc386117934)

[11.13 SearchPatients 21](#_Toc386117935)

[11.14 SearchPatientsResult 21](#_Toc386117936)

[11.15 Error 22](#_Toc386117937)

# Scope

This document is intended to be used for EMR integration tasks. ndd allows interested parties to create their own EMR module in order to connect their system with EasyWarePro (e.g. Easy on-PC). This document explains how it works and what is required to create a customized plug-in.

**Warning:** Please note, that the developer assumes full liability for using their plug-in. The developer must verify functionality, in order to ensure patient safety and proper operation and must act in compliance with the appropriate medical regulations and standards.

# Conventions

## Abbreviations

### EMR , EHR and EPR

The terms EHR (electronic health record), EPR (electronic patient record) and EMR (electronic medical record) are often used interchangeably, although differences between them can be defined. The EMR can, for example, be defined as the patient record created in hospitals and ambulatory environments, and which can serve as a data source for the EHR. It is important to note that an EHR is generated and maintained within an institution, such as a hospital, integrated delivery network, clinic, or physician office, to give patients, physicians and other health care providers, employers, and payers or insurers access to a patient's medical records across facilities (source: http://en.wikipedia.org/wiki/Electronic\_medical\_record)

### EWP

EasyWare Pro, ndd’s spirometry software used in the following ndd products:

Easy on-PC, EasyOne Pro®, EasyOne Pro® LAB.

## Format

An underlined value indicates a default value.

# Introduction

The EMR function of EasyWarePro is extendable through plug-ins. This interface provides programmatic access to EasyWarePro using XML data. Writing your own plug-in allows you build a seamless, integrated experience for the clinician.

EasyWarePro  
(Easy on-PC)

Console

<XML>

HL7

XML files

3rd Party

# Design

In order to allow further development and extensions only a basic interface with few methods exists. These methods must not change as we create newer SW versions. To gain flexibility commands and data are transmitted as xml.

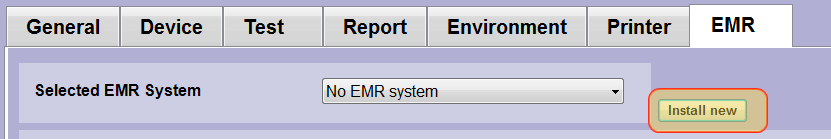
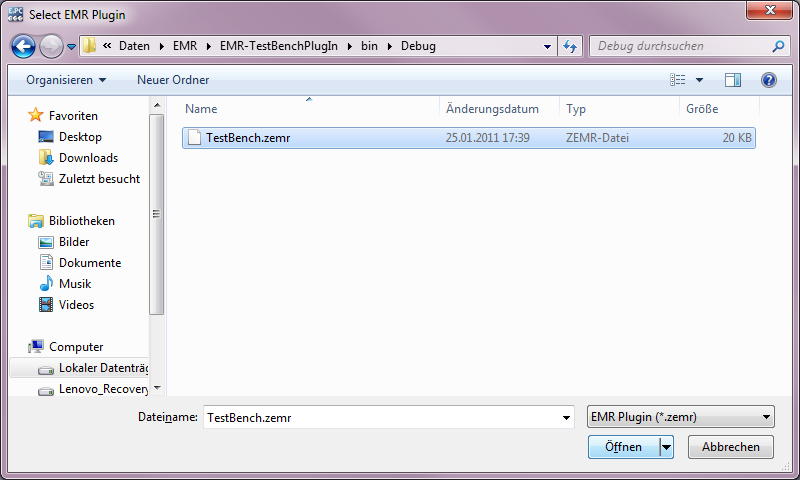
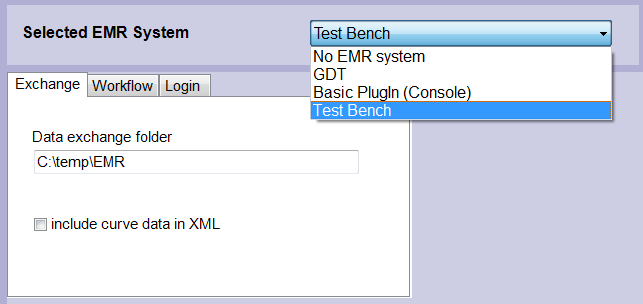
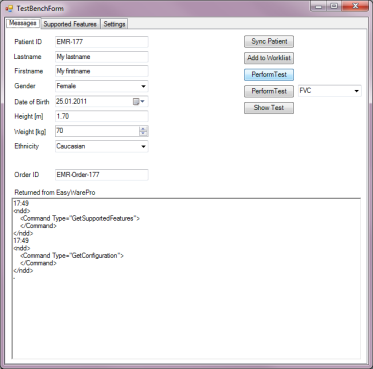
# Requirements/Limitations

Plug-in’s have to be implemented as .NET dll’s.

# How to get started

This chapter will show you how Easy on-PC interacts with EMR Plug-Ins. By installing the Plug-In “Test Bench” you will have a graphical user interface which allows you to manually start EMR procedures and see the result.

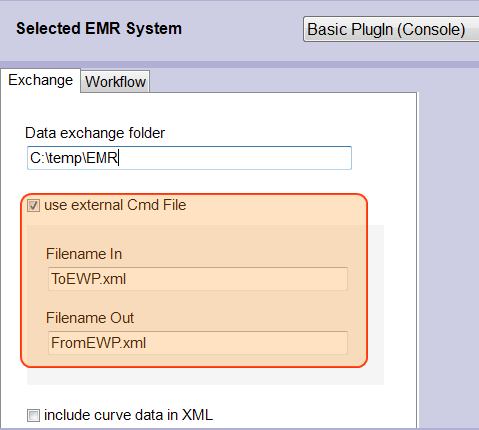
|  |
| --- |
| EasyWarePro  Test Bench |

1. Install Easy on-PC (<http://www.ndd.ch/Downloads/software.aspx>)
2. Install Visual Studio
3. Open EMR-PlugIns.sln within Visual Studio  
   (you can download the SDK from [ftp://nddclient:support79@ftp.ndd.ch/EasyWarePro/EMR/SDK EMR plugin.zip](ftp://nddclient:support79@ftp.ndd.ch/EasyWarePro/EMR/SDK%20EMR%20plugin.zip))
4. Compile the project
5. Zip the dll files in .. \EMR\EMR-TestBenchPlugIn\bin\Release\
6. Change to file extension to ‘.zemr’.
7. In EasyWarePro press Utilities > Configuration > EMR > Install new  
     
     
   Alternatively you could copy the dll files to C:\Program Files (x86)\ndd Medizintechnik\Easy on-PC\Plugins\TestBench
8. Select Test Bench and configure it the way you want. (You can enter some random characters for Login and Password).  
   
9. Enter/Change the patient data and pressPerform Test, for example.

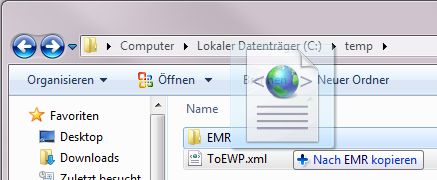
# Control Easy on-PC with your own XML file

The provided plug-In “Basic PlugIn” contains an example how you can send commands to Easy on-PC using your own XML files.

Preparation

1. In EasyWarePro press Utilities > Configuration > EMR
2. Select Basic PlugIn
3. Configure Exchange folder and define the name of the file sent to Easy on-PC and the file you expect from Easy on-PC  
   
4. Press OK

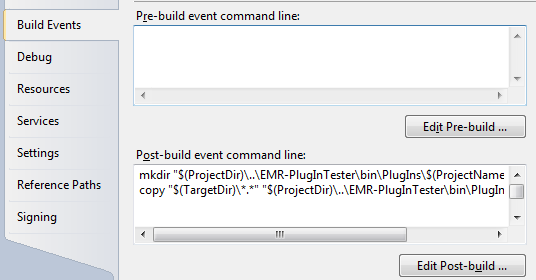
Sending file to Easy on-PC

1. Prepare an xml file according to the rules defined in this document (or copy a file from the output of “Test Bench”).
2. Copy that file to the exchange folder  
   

# Create your own Plug-In

This chapter explains how you can quickly create your own Plug-In and test it with an additional tool that is easier to handle than Easy on-PC.

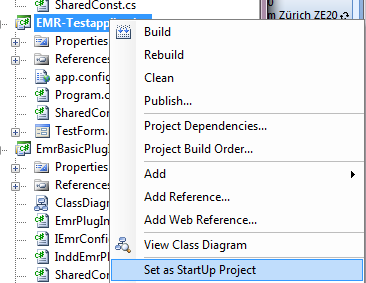
|  |
| --- |
| EMR-TestApplication.exe  Your PlugIn |

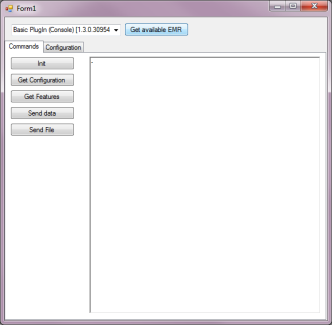
1. Open EMR-PlugIns.sln within Visual Studio
2. Copy and/or rename one of the existing example projects (e.g. EmrPlugInBasic).
3. In project settings make sure it gets copied to the EMR-PlugInTester folder 
4. Go to

|  |
| --- |
| /// <summary>  /// Name shown in the EMR selection drop down list.</summary>  /// <remarks>required</remarks>  public static string Name |

and change the name

1. Adapt the Plug-In to your needs  
   (search for //ToDo: )
2. In Visual Studio select the project EMR-PlugInTester and set it as your StartUp Project



1. Start Debugging (F5)
2. Press “Get available EMR” and select your new created Plug-In   
   

# XML structure

The XML data can contain command information, patient data and test results.

<ndd>

<Command Type="NameOfCommand">

<Parameter Name="NameOfParameter">Value of Parameter</Parameter>

..

</Command>

<Patients>

<Patient ID="xxx">

*.. Testresults*

</Patient>

<Patient ID="yyy">

..

</Patient>

</Patients>

</ndd>

# Interface

## Plug-In interface

In order to be detected and loaded a Plug-in has to implement the following interface.

/// <summary>

/// Interface which identifies EMR-Plug-In. Every Plug-In must implement the the elements of   
/// this interface.</summary>

/// <remarks>required: elements are identified by their names. Don't rename it!</remarks>

public interface InddEmrPlugIn

{

//-- static Info/Configuration --

//static string Name { get; } //module name

//static Control ConfigurationControl //required when Plug in is configurable

//Constructor() //without parameter

//-- Start/Stop --

//start EMR module

bool Initialize(System.Collections.Generic.Dictionary<string, string> configValues);

//stop EMR module

void Close();

//- receiving data –

//handle EasyWarePro messages: xml message (assumption starting with '<' -> xml string, else path to xml file)

string ReceiveXmlMessage(string strXmlMessage);

//- sending data –

//callback to return data as xml string or file

OnEmrSendMessageDelegate SendXmlCallback { set; }

}

public delegate string OnEmrSendMessageDelegate(string strMessage);

## Configuration

If the Plug-in should be configurable within EasyWarePro it has to provide the static propertypublic static Control ConfigurationControl { get; }

That control has to implement the following interface.

/// <summary>

/// A control handling the configuration entry has to implement the following methods.</summary>

interface IUserControlEmrConfig

{

/// <summary>

/// Load previous stored values onto control.</summary>

/// <param name="configValues">list of key-value parameters stored in EasyWarePro</param>

void SetGuiValues(System.Collections.Generic.Dictionary<string, string> configValues);

/// <summary>

/// Validates entries on the form.</summary>

/// <returns><code>true</code> = values are valid</returns>

bool ValidateEntries();

/// <summary>

/// Reads the values on the configuration form.</summary>

/// <returns>list of key-value parameters</returns>

Dictionary<string, string> GetGuiValues();

}

# Commands

The exchanged XML data can contain command information. The following commands are supported.

## GetSupportedFeatures

Requests the list of supported features.

Direction

|  |  |  |
| --- | --- | --- |
| EWP | -> | Plug-In |

EasyWarePro  
(Easy on-PC)

PlugIn

<Command Type="GetSupportedFeatures">

<Command Type="SupportedFeatures">

Parameter

-

## SupportedFeatures

Returns the features which the plug-in supports.

Direction

|  |  |  |
| --- | --- | --- |
| Plug-In | -> | EWP |

Parameter

|  |  |  |  |
| --- | --- | --- | --- |
| Name | value | required | description |
| **SearchPatients** | True/False | - | Flag whether PlugIn supports the SearchPatients command. If True the filter in the patients list view in EWP will be replaced by Search EMR. (see also 11.13 *SearchPatients*) |
| **Worklist** | True/False | - | Flag whether application should be closed after EMR session. |

Example

<ndd>

  <Command Type="SupportedFeatures">

    <Parameter Name="SearchPatients">False </Parameter>

  </Command>

</ndd>

## GetConfiguration

Requests EWP relevant configuration values.

Direction

|  |  |  |
| --- | --- | --- |
| EWP | -> | Plug-In |

EasyWarePro  
(Easy on-PC)

PlugIn

<Command Type="GetConfiguration">

<Command Type="Configuration">

Parameter

-

## Configuration

Returns EWP relevant configuration values. They don’t have to be identical to the list used in Initialize() and UserControlEmrConfig.GetGuiValues().

Directions

|  |  |  |
| --- | --- | --- |
| Plug-In | -> | EWP |

Parameter

|  |  |  |  |
| --- | --- | --- | --- |
| Name | value | required | description |
| **FileExchangeFolder** | string | - | Path where the xml data file should be written to. |
| **SkipPatientSelectionConfirmation** | True/False | - | Flag whether the patient demographic screen (edit patient) is shown for confirmation purposes. (Supported since EWP version >= 1.5.0.18) |
| **CloseAfterTest** | True/False | - | Flag whether application should be closed after EMR session. |
| **IncludeTrialValues** | True/False |  | Flag whether test results should contain the result parameter of each trial (results in more data!). |
| **IncludeCurveData** | True/False | - | Flag whether curve data should be included in the returned test result (results in more data!). Currently only flow and volume curves of spirometry tests are supported. |
| **AttachReport** | True/False | - | Flag whether report (print out) should be attached. |
| **AttachmentFormat** | Enum | - | Default is PDF, the following formats also exist  PDF, CSV, TXT, RTF, XLS, PNG, GIF, TIF, JPG  MHT |
| **AttachmentFileName** | String |  | Following place holders can be used:  %PatientID%  %Firstname%  %Lastname%  %DOB% (EWP version >= 1.3.0.5)  %OrderID%  %TestTime% (EWP version >= 1.8.0.14)  %DeviceSN% (EWP version >= 1.8.0.14)  %Computer% (EWP version >= 1.8.0.14)  Example: Report\_%PatientID%\_%OrderID% |

Additional parameters will be ignored.

## SyncPatient

Adds a new patient to the DB (when that patient ID does not already exist), or else updates patient data (like name).

Directions

|  |  |  |
| --- | --- | --- |
| Plug-In | -> | EWP |

Parameter

-

Example

The XML file will add 3 patients. Missing values is not a problem. The user will be prompted to complete before he starts a new test.

<?xml version="1.0" encoding="utf-16"?>

<ndd>

  <Command Type="SyncPatient">

  </Command>

  <Patients>

    <Patient ID="PSM-11213">

      <LastName>Smith</LastName>

      <FirstName>Peter</FirstName>

      <PatientDataAtPresent>

        <Height>1.82</Height>

        <Weight>80</Weight>

        <Ethnicity>Caucasian</Ethnicity>

        <Smoker>No</Smoker>

        <Gender>Male</Gender>

        <DateOfBirth>1968-11-08</DateOfBirth>

      </PatientDataAtPresent>

    </Patient>

    <Patient ID="TestID1">

      <LastName>Doe</LastName>

      <FirstName>John</FirstName>

      <PatientDataAtPresent>

        <DateOfBirth>1968-11-20</DateOfBirth>

      </PatientDataAtPresent>

    </Patient>

    <Patient ID="TestID2">

    </Patient>

  </Patients>

</ndd>

## AddToWorklist

Adds the patient (whenthat patient ID does not already exist) and the order to the DB. You have to activate the worklist via *SupportedFeatures*.

Directions

|  |  |  |
| --- | --- | --- |
| Plug-In | -> | EWP |

Parameter

The parameters below are the columns of the table which are shown in the orders view. This allows you to pass along information for the technician.

|  |  |  |  |
| --- | --- | --- | --- |
| Name | value | required | description |
| **OrderID** | string | yes | EMR specific identification. Must be unique.  (Supported since EWP version >= 1.8.0.14) |
| **TestType** | string | - | Defines whether FVL, SVC, DLCO, … test should be started (skip test type selection form)  (Supported since EWP version >= 1.8.0.14) |
| **Scheduled** | DateTime | - | Planned/scheduled date  Alternative names:  Expected Date/Time  Visit Date  Service date  (Supported since EWP version >= 1.8.0.14) |
| **OrderedBy** | string |  | Who ordered the test.  (Supported since EWP version >= 1.8.0.14) |
| **Created** | DateTime |  | When the order was placed.  (Supported since EWP version >= 1.8.0.14) |
| **Comment** | string |  | Any comment.  (Supported since EWP version >= 1.8.0.14) |

Example

The XML file will add 3 patients. Missing values is not a problem. The user will be prompted to complete before he starts a new test.

<?xml version="1.0" encoding="utf-16"?>

<ndd>

  <Command Type="AddToWorklist">

     <Parameter Name="OrderID">My Reference abc123</Parameter>

  <Parameter Name=" Scheduled">2010-11-26T11:34:09</Parameter>

     <Parameter Name="TestType">FVC</Parameter>

  </Command>

  <Patients>

    <Patient ID="PSM-11213">

      <LastName>Smith</LastName>

      <FirstName>Peter</FirstName>

      <PatientDataAtPresent>

        <Height>1.82</Height>

        <Ethnicity>Caucasian</Ethnicity>

        <Gender>Male</Gender>

        <DateOfBirth>1968-11-08</DateOfBirth>

      </PatientDataAtPresent>

    </Patient>

  </Patients>

</ndd>

## RemoveWorklist

This command removes the order from the worklist. The action is only performed when the test hasn’t been executed yet.

This command is supported since V. 1.8.0.14.

Directions

|  |  |  |
| --- | --- | --- |
| Plug-In | -> | EWP |

Parameter

|  |  |  |  |
| --- | --- | --- | --- |
| Name | value | required | description |
| **OrderID** | string | yes | EMR specific order identification. Must be unique. |

## PerformTest

This command starts a lung function test. The patient is added or updated.

Directions

|  |  |  |
| --- | --- | --- |
| Plug-In | -> | EWP |

Parameter

|  |  |  |  |
| --- | --- | --- | --- |
| Name | value | required | description |
| **OrderID** | string | - | EMR specific identification. Must be unique. |
| **TestType** | string | - | Defines whether FVL, SVC, DLCO, … test should be started (skips the test type selection form). |

Example

Sending the data below will start a new FVC spirometry test.

<?xml version="1.0" encoding="utf-16"?>

<ndd>

  <Command Type="PerformTest">

    <Parameter Name="TestType">FVC</Parameter>

  </Command>

  <Patients>

    <Patient ID="EMR-177">

      <LastName>My lastname</LastName>

      <FirstName>My firstname</FirstName>

      <PatientDataAtPresent>

        <DateOfBirth>2012-07-13</DateOfBirth>

        <Gender>Female</Gender>

        <Height>1.7</Height>

        <Ethnicity>Caucasian</Ethnicity>

      </PatientDataAtPresent>

    </Patient>

  </Patients>

</ndd>

## TestResults

Returns test results (either when requested through GetTestResults or when Test has been performed).

Directions

|  |  |  |
| --- | --- | --- |
| EWP | -> | Plug-In |

Parameter

|  |  |  |  |
| --- | --- | --- | --- |
| Name | value | required | description |
| **Attachment** | path | n/a | Path to the attachment file (e.g. PDF).  (Attachment can be configured, see ) |

## GetTestResults

Request the results of the specified test.

Directions

|  |  |  |
| --- | --- | --- |
| Plug-In | -> | EWP |

Parameter

|  |  |  |  |
| --- | --- | --- | --- |
| Name | value | required | description |
| **DateTime** | date | yes | Identifies the test of the given patient (needs to be exact to the second). Can be left empty if given patient has only 1 test. |

## ReviewTest

Shows an existing test.

Directions

|  |  |  |
| --- | --- | --- |
| Plug-In | -> | EWP |

Parameter

|  |  |  |  |
| --- | --- | --- | --- |
| Name | value | required | description |
| **OrderID** | string | \* | EMR specific identification |
| **DateTime** | date | \* | Time the test has been performed |

\* Either OrderID or DateTime is required.

Example

Sending the data below will open the test performed at the time provided. In case DateTime is not provided or no test exists at that time, then the history of that patient is shown.

<?xml version="1.0" encoding="utf-16"?>

<ndd>

  <Command Type="ReviewTest">

    <Parameter Name="DateTime">2010-11-26T11:34:09</Parameter>

  </Command>

  <Patients>

    <Patient ID="PSM-11213">

      <LastName>Smith</LastName>

      <FirstName>Peter</FirstName>

    </Patient>

  </Patients>

</ndd>

## PrintTest

Opens the print preview or prints an existing test.

Directions

|  |  |  |
| --- | --- | --- |
| Plug-In | -> | EWP |

Parameter

|  |  |  |  |
| --- | --- | --- | --- |
| Name | value | required | description |
| **OrderID** | string | \* | EMR specific identification |
| **DateTime** | date | \* | Time the test has been performed |
| **ShowPreview** | True/False | - | True: print preview is shown  False: report is sent to printer. |
| **ReportLayout** | string | - | Defines the layout to be used. |

\* Either OrderID or DateTime is required.

Example

The command below will print the test performed at the time provided.

<?xml version="1.0" encoding="utf-16"?>

<ndd>

<Command Type="PrintTest">

<Parameter Name="DateTime">2013-08-07T09:48:00.184</Parameter>

<Parameter Name="ShowPreview">False</Parameter>

<Parameter Name="ReportLayout">BuildInReport::Summary</Parameter>

</Command>

<Patients>

<Patient ID="My PatientID" >

<LastName>My Lastname</LastName>

<FirstName>My Firstname</FirstName>

</Patient>

</Patients>

</ndd>

The next example defines a custom report layout and opens the print preview.

<?xml version="1.0" encoding="utf-16"?>

<ndd>

<Command Type="PrintTest">

<Parameter Name="DateTime">2013-08-07T09:48:00.184</Parameter>

<Parameter Name="ShowPreview">True</Parameter>

<Parameter Name="ReportLayout">YourPersonalReportLayout.repx</Parameter>

</Command>

<Patients>

<Patient ID="My PatientID">

<LastName>My Lastname</LastName>

<FirstName>My Firstname</FirstName>

</Patient>

</Patients>

</ndd>

## SearchPatients

Requests a list of patients that match the search query.

Directions

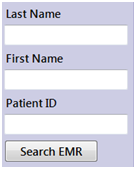
|  |  |  |
| --- | --- | --- |
| EWP | -> | Plug-In |

EasyWarePro  
(Easy on-PC)

PlugIn

<Command Type="SearchPatients">

<Command Type="SearchPatientsResult">



Parameter

|  |  |  |  |
| --- | --- | --- | --- |
| Name | value | required | description |
| **FirstName** | string | - | Part of or full first name |
| **LastName** | string | - | Part of or full last name |
| **PatientID** | string | - | Part of or full patient ID |

Example

<ndd>

<Command Type="SearchPatients">

<Parameter Name="FirstName">J</Parameter>

<Parameter Name="LastName">Due</Parameter>

</Command>

</ndd>

## SearchPatientsResult

Result of search query.

Directions

|  |  |  |
| --- | --- | --- |
| Plug-In | -> | EWP |

Parameter

-

## Error

Indicates that an error occurred.

Directions

|  |  |  |
| --- | --- | --- |
| EWP | <-/-> | Plug-In |

Parameter

|  |  |  |  |
| --- | --- | --- | --- |
| Name | value | required | description |
| **Message** | String | - | Error message |
| **Details** | String | - | More detailed error message (e.g. technical) |