

XPages und Domino App Dev Erweiterungen

Philippe Riand, IBM
Niklas Heidloff, IBM



Disclaimer

- The information contained in this publication is provided for informational purposes only. While efforts were made to verify the completeness and accuracy of the information contained in this publication, it is provided AS IS without warranty of any kind, express or implied. In addition, this information is based on IBM's current product plans and strategy, which are subject to change by IBM without notice. IBM shall not be responsible for any damages arising out of the use of, or otherwise related to, this publication or any other materials. Nothing contained in this publication is intended to, nor shall have the effect of, creating any warranties or representations from IBM or its suppliers or licensors, or altering the terms and conditions of the applicable license agreement governing the use of IBM software.
- References in this publication to IBM products, programs, or services do not imply that they will be available in all countries in which IBM operates. Product release dates and/or capabilities referenced in this presentation may change at any time at IBM's sole discretion based on market opportunities or other factors, and are not intended to be a commitment to future product or feature availability in any way. Nothing contained in these materials is intended to, nor shall have the effect of, stating or implying that any activities undertaken by you will result in any specific sales, revenue growth or other results.
- All customer examples described are presented as illustrations of how those customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics may vary by customer.
- IBM, the IBM logo, Lotus, Lotus Notes, Notes, Domino, DeveloperWorks and Workplace are trademarks of International Business Machines Corporation in the United States, other countries, or both.
- Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both
- All references to Renovations refer to a fictitious company and are used for illustration purposes only.
- Other company, product, or service names may be trademarks or service marks of others.

Agenda

- Access to relational Databases from XPages
- JDBC Driver to access NSFs
- Domino REST APIs
- Workflow for XPages
- OSGi Tasklet Service

Agenda

- Access to relational Databases from XPages
- JDBC Driver to access NSFs
- Domino REST APIs
- Workflow for XPages
- OSGi Tasklet Service

XPages Extension Library OpenNTF


- Available at <http://extlib.openntf.org/>
 - Active conversation with developers in the Discussions and Defects section

[Overview](#)
[Downloads / Releases](#)
[News](#)
[Screenshots](#)
[Documentation](#)
[Feature Requests](#)
[Defects](#)
[Discussions](#)
[Help Wanted](#)
[Reviews](#)

[Email Project Owners](#)
[Create Feature Request](#)
[Create Defect](#)
[Create Discussion](#)
[Create Review](#)
[Rate Project](#)
[Subscribe to Project \(RSS\)](#)

XPages Extension Library

Owners	Philippe Riand , Niklas Heidloff , Dave Delay , Akihiro Kosugi , Maire Kehoe , Tony McGuckin , John Piermarini , Martin Donnelly , James Quill , Darin Egan , Padraic Edwards , Andrejus Chaliapinas , Lorcan McDonald , Simon McLoughlin	Category	Software Development Tools
Contributors	-	Platform	R 8.5.3
Downloads	22769 Download latest release	Last Release	19 Nov 2011
Rating	★★★★★ (21 ratings)	Project Creation	1 Sep 2010
Status	Active	Short URL	http://extlib.openntf.org
Description	XPages Extension Library	In Catalog	No



XPages Extension Library OpenNTF

- There are 2 sets of plugins included in OpenNTF delivery
 - “**extlib**” and “**extlibx**”.
- Each set is wrapped into its own **XPages specific library**
 - com.ibm.xsp.extlib.library
 - com.ibm.xsp.extlibx.library
- The “**extlibx**” library contains eXperimental code
 - RDBMS support
 - Social Enabler and others
- Experimental code can get added to the “extlib” library
 - once the code has matured and there is sufficient customer demand

XPages Extension Library OpenNTF

- XPages apps can depend on both the extlib and extlibx libraries
 - RDBMS access requires both dependencies

Advanced Properties

XPage Libraries

Select the libraries of extended XPage controls to use in this application.

Library ID	
<input checked="" type="checkbox"/>	com.ibm.xsp.extlib.library
<input checked="" type="checkbox"/>	com.ibm.xsp.extlibx.library
<input type="checkbox"/>	
<input type="checkbox"/>	
<input type="checkbox"/>	

When running on the Web, the libraries must be available on the server. When running on the Notes client, the library plug-ins must be installed on the client.

XPAGES Extension Library Demo App

- XPAGESJDBC.nsf
 - SQL table data and sorting
 - SQL query data
 - SQL query with count and page navigation
 - SQL query data when query is in a file
 - SQL parameterized query



DEMO

XPAGES Home

XPages Relational Data Access Demo

Home

Static View Panel - Table

Static View Panel - Query

Static View Panel - Query+Count

Static View Panel - SQL File

Static View Panel - Parameter

Dynamic View Panel

Cached RowSet Read/Write data source

SQL Query

Database Meta Data

Table Meta Data

@Functions()

SQL table

JDBC query data source displaying a table within a view panel. The 'calculateCount' property is set to true.

Previous 1 2 3 4 5 6 7 8 ... Next

ID	First Name	Last Name	City	State
0	Lorraine	Cain	Las Vegas	NV
1	Willie	Byers	Philadelphia	PA
10	Floyd	Lane	Portland	OR
100	Debra	Brewer	Huntington Beach	CA
101	Hope	Gould	Clearwater	FL
102	Alex	Barlett	Provo	UT
103	Andrew	Villareal	Evansville	IN
104	Leigh	Page	Stamford	CT
105	Andy	Alston	Murfreesboro	TN
106	Lolita	Logan	Alexandria	VA

(c) IBM 2010

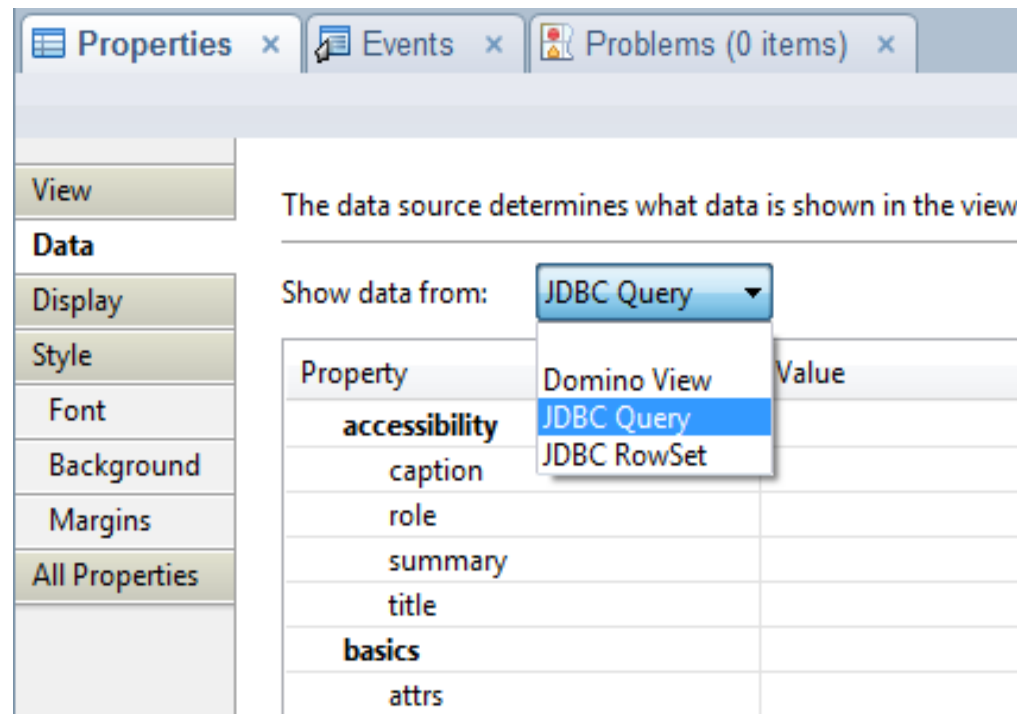
New XPages Data Sources for RDBMS

- XPages leverages JDBC when connecting to relational databases
 - Accesses any database with a JDBC driver available
 - You could access
 - DB2,
 - Oracle,
 - MS SQL,
 - MySQL,
 - Derby
 - ... you name it!
 - You need to obtain the JDBC driver from the database vendor or third party provider
- RDBMS data is accessed directly – no synchronization is performed with NSF
- XPages efficiently manages the connections through a connections pool
- Uses NSF specific connections
- The data is accessed using a set of dedicated data sources

New XPages Data Sources for RDBMS

- extlibx library adds 2 new data sources:
 - JDBC Query for Read Only access
 - JDBC RowSet for CRUD (create/read/update/delete) access.
 - Current implementation is based on `com.sun.rowset.CachedRowSetImpl`,
 - Could be configured to use your own via the `rowSetJavaClass` property

- Each of the provided new data sources could be easily utilized either by:
 - the core View Container Control (`xp:viewPanel` element)
 - the XPage itself (`xp:view` element)
- The new data sources are now available in the Domino Designer Property Panel



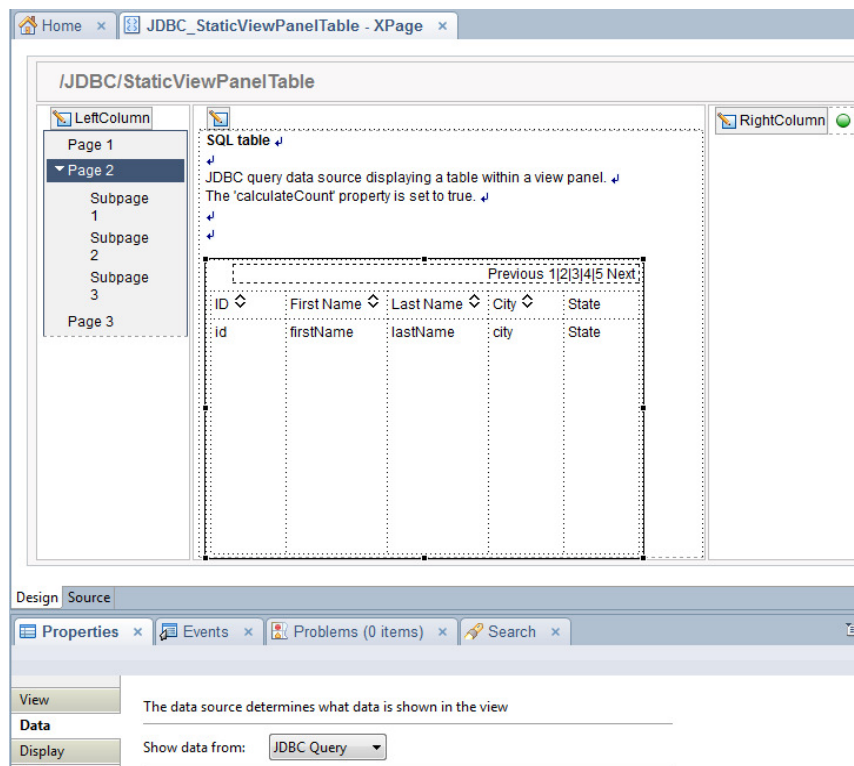
New XPages Data Sources for RDBMS

- For both JDBC Query and JDBC RowSet data sources, access to fields in SQL data are:
 - via the core xp:viewColumn element
 - using its columnName property
 - i.e. the same way you do with Domino View data source
- Currently you need to know what RDBMS fields are returned in your specified SQL table or by a SQL query
- Code snippet showing the definition of a view column using the “firstName” RDBMS field
 - The column can even be made sortable!

```
<xp:viewColumn id="viewColumn3" columnName="firstName">
  <xp:this.facets>
    <xp:viewColumnHeader xp:key="header"
      id="viewColumnHeader3" value="First Name" sortable="true">
    </xp:viewColumnHeader>
  </xp:this.facets>
</xp:viewColumn>
```

New XPages Data Sources for RDBMS

- It's very easy to bring your relational data into an XPages enabled web application using familiar Domino Designer controls



SQL table

JDBC query data source displaying a table within a view panel.
The 'calculateCount' property is set to true.

Previous 1 2 3 4 5 6 7 8 ... Next				
ID	First Name	Last Name	City	State
0	Lorraine	Cain	Las Vegas	NV
1	Willa	Byers	Philadelphia	PA
10	Floyd	Lane	Portland	OR
100	Debra	Brewer	Huntington Beach	CA
101	Hope	Gould	Clearwater	DL
102	Alex	Bartlett	Provo	UT
103	Andrew	Villarreal	Evansville	IN
104	Leigh	Page	Stamford	CT
105	Andy	Alston	Murfreesboro	TN
106	Lolita	Logan	Alexandria	VA

New XPages Data Sources for RDBMS

- For the CRUD scenario – you could easily develop a record editing dialog, taking values from the JDBC RowSet data source

Read/Write data source

The runtime is featuring a new data source leveraging a standard JDBC CachedRowSet. This data source can be displayed using a regular view panel, but its records can also be edited and then committed back to the RDBMS. When you click on a record link, then a popup dialog allows you to modify the record, while a click on the 'insert' button creates a new record. But the changes are not committed to the database unless the save() action is triggered. This data source also displays the deleted rows, using the showDeleted property. Be careful when using this data source as all the records, up to the maxRows property value, are extracted from the database and kept as part of the data source. You must then use this data source with limited data sets.



Previous 1 2 3 4 5 6 7 8 ... Next					
	ID	First Name	Last Name	City	State
<input type="checkbox"/>	0	Lorraine	Cain	Las Vegas	NV
<input checked="" type="checkbox"/>	1	Willa	Byers	Philadelphia	PA
<input type="checkbox"/>	2	Joanna	Wooten	West Jordan	UT
<input type="checkbox"/>	3	Gwen	Fulton	Jacksonville	DL
<input type="checkbox"/>	4	Nathan	Harper	Salt Lake	UT
<input type="checkbox"/>	5	Nichole	Hopper	Houston	TX
<input type="checkbox"/>	6	Janet	House	El Paso	TX
<input type="checkbox"/>	7	Marina	Strickland	Port St. Lucie	DL
<input type="checkbox"/>	8	Danielle	Morales	Fort Worth	TX
<input type="checkbox"/>	9	Susie	Chandler	Kansas	MO
Save Changes Insert New Record Delete Selected Records					

User Record

User Record

Enter here the data for the user

'Id:

1

'First Name:

Willa

'Last Name:

Byers

'City:

Philadelphia

'State:

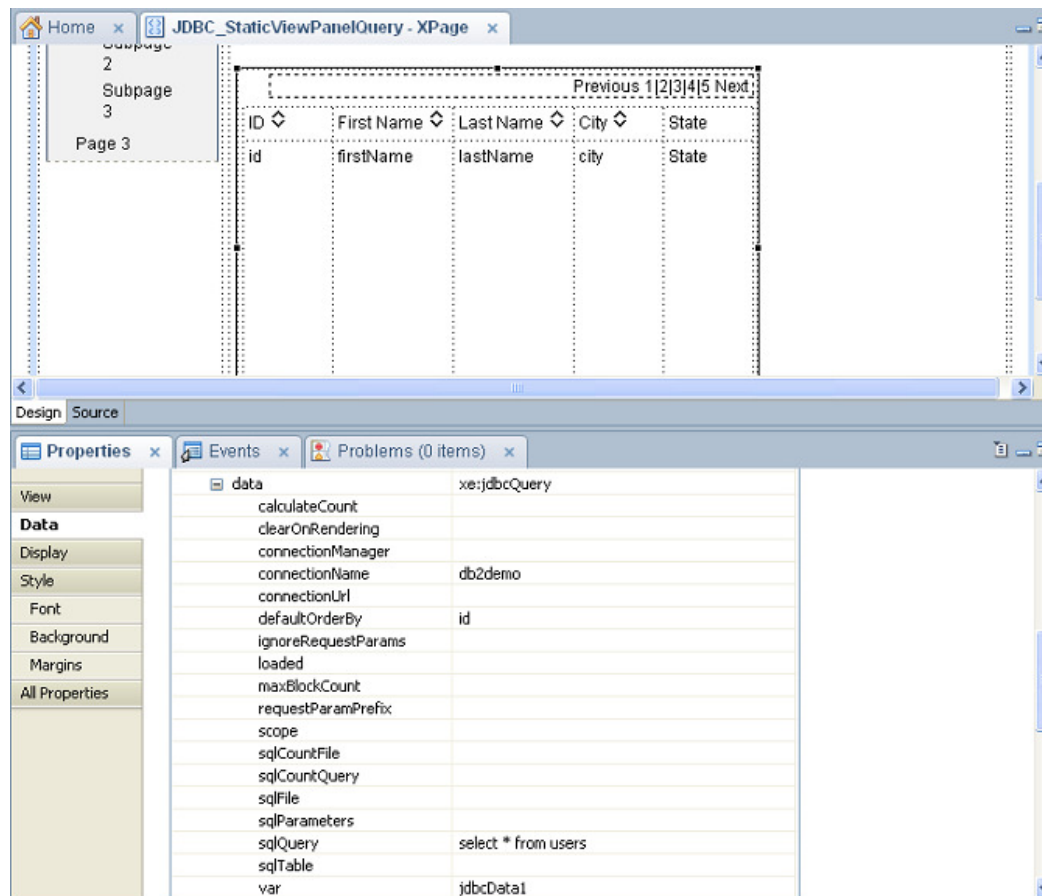
PA

Ok

Cancel

XPages Extension Library Demo App

- DEMO of the Domino Designer view control configuration, with the new JDBC data sources:
 - JDBC Query usage
 - JDBC RowSet usage



Making RDBMS Connections

- To connect to your RDBMS you need to create a special file resource

- .jdbc extension (for example “db2test.jdbc”)
 - Specifies the JDBC driver class to be used
 - The database host/port/name and users' credentials

<jdbc>

```
<driver>com.ibm.db2.jcc.DB2Driver</driver>
```

```
<url>jdbc:db2://testvm:50000/SAMPLE</url>
```

```
<user>demo</user>
```

```
<password>test123</password>
```

</jdbc>

- The file name before the “.jdbc” extension is the connection name (in this case “db2test”)
 - Should be used in the JDBC data sources configuration
- File resources with the “.jdbc” extension
 - Should be placed within an XPages application (NSF) under the WebContent/WEB-INF/jdbc folder
 - This folder is visible within Domino Designer in the Navigator/Package Explorer views only.
 - It is not visible in the default Applications view
- File resources within the WebContent/WEB-INF
 - Cannot be accessed via URL syntax
 - Your password and connection data is secure

Making RDBMS Connections

- SQL data can be made available by using the following properties:
 - sqlTable (the runtime will generate a select * from <sqlTable>)
 - sqlQuery (SQL query, which could be either static or computed)
 - sqlFile (file resource with the “.sql” extension, containing the SQL statement)
- The Core Pager control (xp:pager) can be used to navigate through the SQL data and additional properties can help display the actual number of pages for the JDBC Query data source:
 - calculateCount (false by default, but when set to true, the runtime will generate select count(*) for the corresponding table or query)
 - sqlCountQuery (allows you to specify your own count query)
 - sqlCountFile (file resources with the “.sql” extension, containing the SQL count query)
- For the JDBC RowSet data source, you could limit the number of rows retrieved by:
 - maxRows (default is 0, all rows retrieved, so be careful here)
 - Pager control is already aware of the actual number of pages
- File resources with the “.sql” extension should be placed in an XPages application, under the WebContent/WEB-INF/jdbc folder.

Making RDBMS Connections

- SQL query could be parameterized using the '?' syntax
 - select * from users where state=?
- Each corresponding parameter can be defined using the sqlParameters data source property and this can also be computed

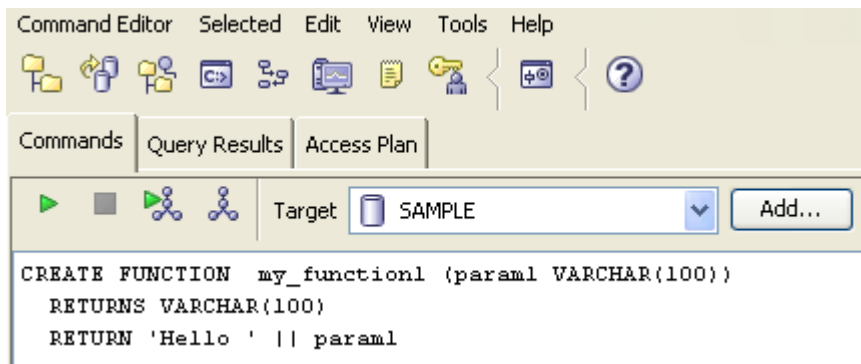
sqlParameters	
sqlParameter [0]	
loaded	
value	# viewScope.state ""
sqlQuery	select * from users where state=?
sqlTable	



```
<xp:this.data>
  <xe:jdbcQuery connectionName="derby1"
    var="jdbcData1" sqlQuery="select * from users where state=?"
    defaultOrderBy="id" calculateCount="true">
    <xe:this.sqlParameters>
      <xe:sqlParameter>
        <xe:this.value><![CDATA[#{javascript:viewScope.state||""}]]></xe:this.value>
      </xe:sqlParameter>
    </xe:this.sqlParameters>
  </xe:jdbcQuery>
</xp:this.data>
```

Making RDBMS Connections

- For your SQL queries and DB2 RDBMS you can:
 - Call User Defined Functions (UDF) using the special **sysibm.dual** table
 - Refer query fields by using aliases



The screenshot shows a 'Command Editor' window with a menu bar (Command Editor, Selected, Edit, View, Tools, Help) and a toolbar. Below the toolbar are tabs for 'Commands', 'Query Results', and 'Access Plan'. A 'Target' dropdown menu is set to 'SAMPLE' with an 'Add...' button. The main text area contains the following SQL code:

```
CREATE FUNCTION my_function1 (param1 VARCHAR(100))  
  RETURNS VARCHAR(100)  
  RETURN 'Hello ' || param1
```



```
<xp:viewPanel rows="10" id="viewPanel1" var="row">  
  <xp:this.facets>  
    <xp:pager partialRefresh="true" layout="Previous Group Next"  
      xp:key="headerPager" id="pager1">  
    </xp:pager>  
  </xp:this.facets>  
  <xp:this.data>  
    <xe:jdbcQuery connectionName="db2test"  
      sqlQuery="select my_function1('test') as col1 from sysibm.dual"  
      var="jdbcData1">  
    </xe:jdbcQuery>  
  </xp:this.data>  
  <xp:viewColumn id="viewColumn1" columnName="col1">  
    <xp:viewColumnHeader value="col1"  
      id="viewColumnHeader1"></xp:viewColumnHeader>  
  </xp:viewColumn>  
</xp:viewPanel>
```



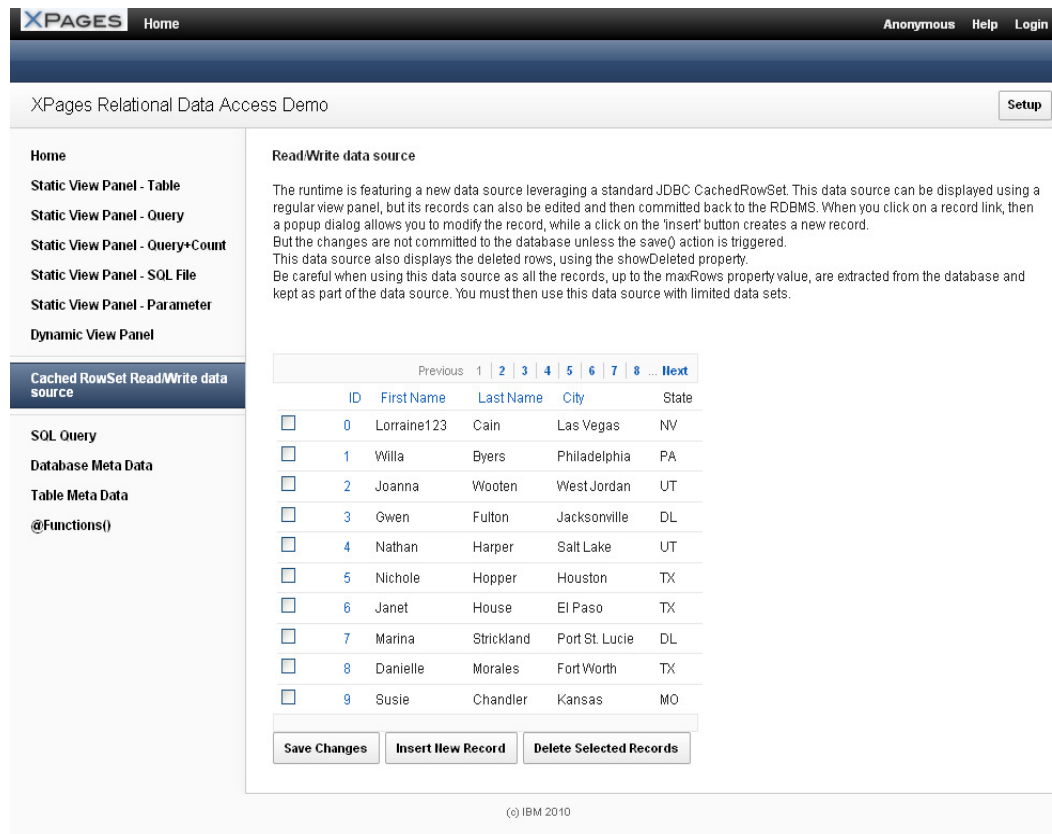
Previous	1	Next
----------	---	------

col1

Hello test

Making RDBMS Connections

- The use of these various properties are demonstrated in the included XPagesJDBC.nsf demo application



XPagesJDBC.nsf Demo Application Interface

Read/Write data source

The runtime is featuring a new data source leveraging a standard JDBC CachedRowSet. This data source can be displayed using a regular view panel, but its records can also be edited and then committed back to the RDBMS. When you click on a record link, then a popup dialog allows you to modify the record, while a click on the 'insert' button creates a new record. But the changes are not committed to the database unless the save() action is triggered. This data source also displays the deleted rows, using the showDeleted property. Be careful when using this data source as all the records, up to the maxRows property value, are extracted from the database and kept as part of the data source. You must then use this data source with limited data sets.

ID	First Name	Last Name	City	State
0	Lorraine123	Cain	Las Vegas	NV
1	Willa	Byers	Philadelphia	PA
2	Joanna	Wooten	West Jordan	UT
3	Owen	Fulton	Jacksonville	DL
4	Nathan	Harper	Salt Lake	UT
5	Nichole	Hopper	Houston	TX
6	Janet	House	El Paso	TX
7	Marina	Strickland	Port St. Lucie	DL
8	Danielle	Morales	Fort Worth	TX
9	Susie	Chandler	Kansas	MO

Buttons: Save Changes, Insert New Record, Delete Selected Records



- DEMO of the additional XPagesJDBC.nsf application capabilities:
 - Programming aspects
 - @Functions usage

Server Side JavaScript APIs and @Functions

- Currently implemented @Functions:
 - @JdbcInsert
 - @JdbcUpdate
 - @JdbcDelete
 - @JdbcExecuteQuery
 - @JdbcDbColumn
 - @JdbcGetConnection
- Used to simplify various RDBMS operations
- Allows programmatic access to RDBMS data

```
// @JdbcInsert
var v = {
    id: id,
    firstName: "Albert",
    lastName: "Einstein",
    city: "Boston",
    state: "MA"
}
var c = @JdbcInsert("db2test","users",v)
// @JdbcUpdate
var v = {
    city: "Boston",
    state: "MA"
}
var c = @JdbcUpdate("db2test","users",v,"id=?",id)
// @JdbcDelete
var c = @JdbcDelete("db2test","users","id=?",id)
// @JdbcExecuteQuery
var sql = "select * from "+tb;
var rs = @JdbcExecuteQuery('db2test',sql);
// @JdbcDbColumn
@JdbcDbColumn("db2test","states","state")
```


Server Side JavaScript APIs and @Functions


- `@JdbcInsert` function has now an additional calling prototype, which allows you to take advantage of the RDBMS table autoincrement/identity field:

```
CREATE TABLE dept(  
  id INTEGER  
    GENERATED ALWAYS AS IDENTITY  
    PRIMARY KEY,  
  name VARCHAR(20)  
);
```



```
<xp:this.action><![CDATA[#{javascript:  
  try {  
  
    var v = {  
      name: "R&D Department"  
    }  
    var idCol = ["id"]  
    var id = @JdbcInsert("db2demo", "Dept", v, idCol)  
    @InfoMessage("Record inserted, id generated: "+id)  
  
  } catch(e) {  
    @ErrorMessage(e);  
  }  
}]]></xp:this.action>
```



 Please check the following:

- Record inserted, id generated: 2

Server Side JavaScript APIs and @Functions

- `@JdbcExecuteQuery` can be used for quick RDBMS data retrieval and additional processing inside your XPage

```
<xp:this.value><![CDATA[#{javascript:
    var sql = "SELECT DEPTNAME FROM DEMO.\"DEPARTMENT\" WHERE DEPTNO LIKE ' " +
        emp.getColumnValue("workdept") + "'";
    var res=@JdbcExecuteQuery("db2test",sql);
    var deptname;
    while (res.next()) {
        // Get the data from the row using the column name
        deptname = res.getString("DEPTNAME");
        // Do some additional processing
    }
    return deptname;
}]]></xp:this.value>
```

- `@JdbcDbColumn` can be used to populate, for example, an XPages combobox control, with the RDBMS table fields' values

```
<xp:selectItems>
    <xp:this.value><![CDATA[#{javascript:@JdbcDbColumn("db2demo","states","state")
    }]]></xp:this.value>
</xp:selectItems>
```

Server Side JavaScript APIs and @Functions

- To access a row in the JDBC data source, inside an `xp:viewPanel`, you need:
 - an index variable specified via the **indexVar** property, for example, 'idx'
 - usage of the data source **getRow(int index)** method, for example, `{ javascript: jdbcData1.getRow (idx) }`, for some event processing
- Changes to the JDBC RowSet data source could be saved back to the RDBMS using:
 - `{ javascript: jdbcData1.acceptChanges() }` syntax

```
<xp:viewPanel rows="10" id="viewPanel1" var="user"
  value="{jdbcData1}" indexVar="idx">
```



```
<xe:dialog id="userDialog" title="User Record">
  <xp:panel>
    <xp:this.data>
      <xe:objectData var="row"
        saveObject="{javascript:jdbcData1.saveRow(row)}">
        <xe:this.createObject><![CDATA[{javascript:var p = param.row
return p ? jdbcData1.getRow(parseInt(p)) : jdbcData1.newRow(getComponent('viewPanel1').getFirst())}]]></xe:this.createObject>
      </xe:objectData>
    </xp:this.data>
    ...
  </xp:panel>
</xe:dialog>
```

Setup of Server/Development Environments

- If you have Upgrade Pack 1 previously installed – you need to uninstall it
- To consume RDBMS data, you need to install the OpenNTF version of the XPages Extension Library,
- Since the OpenNTF version doesn't have an installer, like the UP1 installer, you need to follow these steps:
 - Download the latest ExtensionLibraryOpenNTF-853-YYYYMMDD-HHMM.zip file (it's big!)
 - Unzip that big zip updateSiteOpenNTF.zip file (to be installed on the Domino server/Notes preview server)
 - Unzip all the content in the updateSiteOpenNTF.zip file, so you can access the site.xml, as well as the features/ plugins folders
 - Unzip, from the big zip, the updateSiteOpenNTF-designer.zip file (to be installed in Domino Designer)
 - No need to unzip the updateSiteOpenNTF-designer.zip content

Setup of server/development environments

- Installation on the Domino server should be done using the [XPages Extension Library Deployment in Domino 8.5.3 and IBM XWork Server](#) technique described in the Lotus Notes and Domino Application Development wiki:
 - Involves creation of a new NSF application based on the **updatesite.ntf** template and import of the extracted updateSiteOpenNTF.zip content, using its **site.xml** file
 - Requires specification of an **OSGI_HTTP_DYNAMIC_BUNDLES** variable inside your server's **notes.ini** file to point to the update sites' NSF
 - Uses additional safeguards when loading plugins (ACL)
- Installation for Notes web preview (where no Domino server is running) should be done based on the [Using an XPages Library in Designer](#) technique, which is also described in the Lotus Notes and Domino Application Development wiki:
 - Requires the features/plugins folder content, which was previously extracted from updateSiteOpenNTF.zip, to be merged with the existing features/plugins folders found under the **<Notes_Data>\domino\workspace\applications\eclipse** folder
 - Special attention should be taken not to use, for Notes web preview purpose, another similar looking folder **<Notes_Data>\workspace\applications\eclipse**, whose purpose is to provide features/plugins for the Notes client/Domino Designer functionality

Setup of server/development environments

- In order for the Domino server to load these installed plugins, you need to restart your Domino server, issuing the console command **restart task http**
- To verify the plugins for the extlib and extlibx XPages libraries are properly installed - you should use the OSGI command on the server console:
tell http osgi ss com.ibm.xsp.extlib

```
> tell http osgi ss com.ibm.xsp.extlib
[0A44:0002-0A40] 11/13/2011 08:33:47 PM Framework is launched.
[0A44:0002-0A40] 11/13/2011 08:33:47 PM id      State      Bundle
[0A44:0002-0A40] 11/13/2011 08:33:47 PM 108    <<LAZY>>   com.ibm.xsp.extlib.c
ontrols_8.5.3.20111027-1245
[0A44:0002-0A40] 11/13/2011 08:33:47 PM 109    <<LAZY>>   com.ibm.xsp.extlib.c
ore_8.5.3.20111027-1245
[0A44:0002-0A40] 11/13/2011 08:33:47 PM 110    <<LAZY>>   com.ibm.xsp.extlib.d
omino_8.5.3.20111027-1245
[0A44:0002-0A40] 11/13/2011 08:33:47 PM 111    <<LAZY>>   com.ibm.xsp.extlib.m
obile_8.5.3.20111027-1245
[0A44:0002-0A40] 11/13/2011 08:33:47 PM 112    <<LAZY>>   com.ibm.xsp.extlib.o
neui_8.5.3.20111027-1245
[0A44:0002-0A40] 11/13/2011 08:33:47 PM 113    <<LAZY>>   com.ibm.xsp.extlibx.
core_8.5.3.20111027-1245
[0A44:0002-0A40] 11/13/2011 08:33:47 PM 114    RESOLVED   com.ibm.xsp.extlibx.
oauth_8.5.3.20111027-1245
[0A44:0002-0A40] 11/13/2011 08:33:47 PM 115    RESOLVED   com.ibm.xsp.extlibx.
relational.derby_8.5.3.20111027-1245
[0A44:0002-0A40] 11/13/2011 08:33:47 PM 116    <<LAZY>>   com.ibm.xsp.extlibx.
relational_8.5.3.20111027-1245
[0A44:0002-0A40] 11/13/2011 08:33:47 PM 117    <<LAZY>>   com.ibm.xsp.extlibx.
sbt_8.5.3.20111027-1245
[0A44:0002-0A40] 11/13/2011 08:33:47 PM 118    <<LAZY>>   com.ibm.xsp.extlibx.
8.5.3.20111027-1245
[0A44:0002-0A40] 11/13/2011 08:33:47 PM 119    <<LAZY>>   com.ibm.xsp.extlib_8
.5.3.20111027-1245
```

Setup of server/development environments

- An additional plugin/update site, which wraps the JDBC driver, is required for the installation to connect to a particular RDBMS

```
<plugin>
  <extension point="com.ibm.commons.Extension">
    <service class="extlib.jdbc.db2.DB2DriverProvider"
      type="com.ibm.common.jdbcprovider">
    </service>
  </extension>
</plugin>
```

- you could manually customize a similar JDBC drivers' plugin/update site in the Eclipse IDE, using the extension point **com.ibm.commons.Extension** and a service type of **com.ibm.common.jdbcprovider**
- Your Java class should instantiate the required JDBC driver

- a sample update site for the DB2 JDBC driver is available on the OpenNTF web site for demonstration. Verification on the server, is via the command:
tell http osgi ss extlib.driver.db2

```
package extlib.jdbc.db2;

import java.sql.Driver;
import java.sql.SQLException;

import com.ibm.commons.jdbc.drivers.IJDBCDriverAlias;
import com.ibm.commons.jdbc.drivers.JDBCProvider;

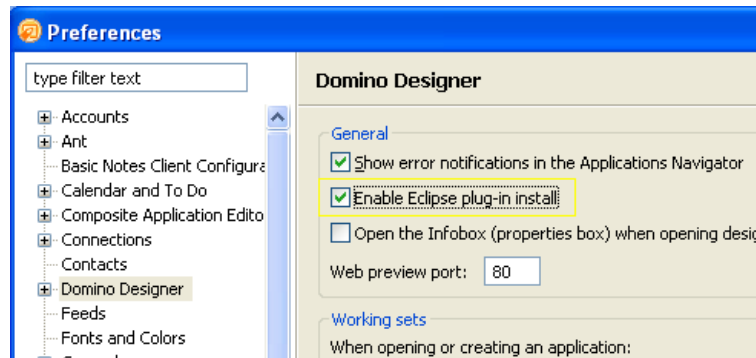
public class DB2DriverProvider implements JDBCProvider {

    public Driver loadDriver(String className) throws SQLException {
        if(className.equals(com.ibm.db2.jcc.DB2Driver.class.getName())) {
            return new com.ibm.db2.jcc.DB2Driver();
        }
        return null;
    }

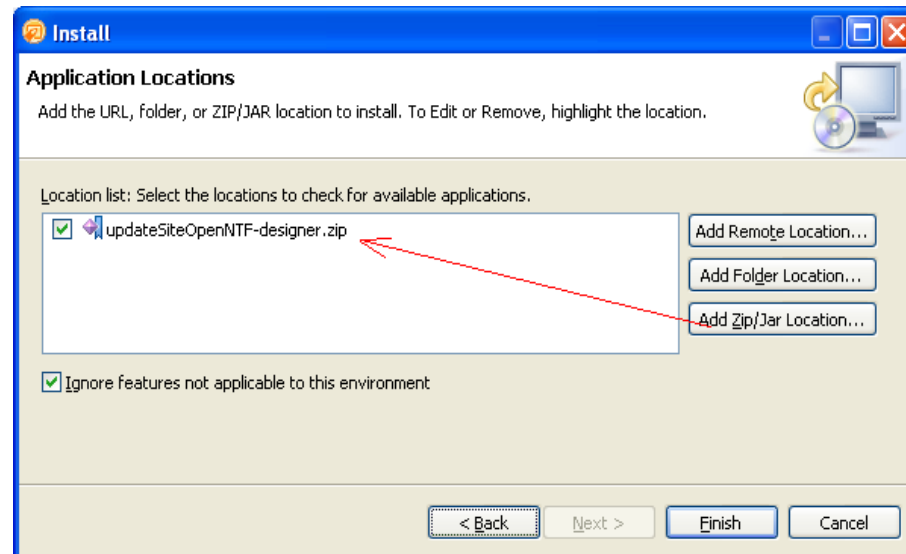
    public IJDBCDriverAlias[] getDriverAliases() {
        return null;
    }
}
```

Setup of server/development environments

- Domino Designer requires the **Enable Eclipse plug-in install** preference to be enabled for the File → Application → Install menu to appear, so that the update site can be manually installed

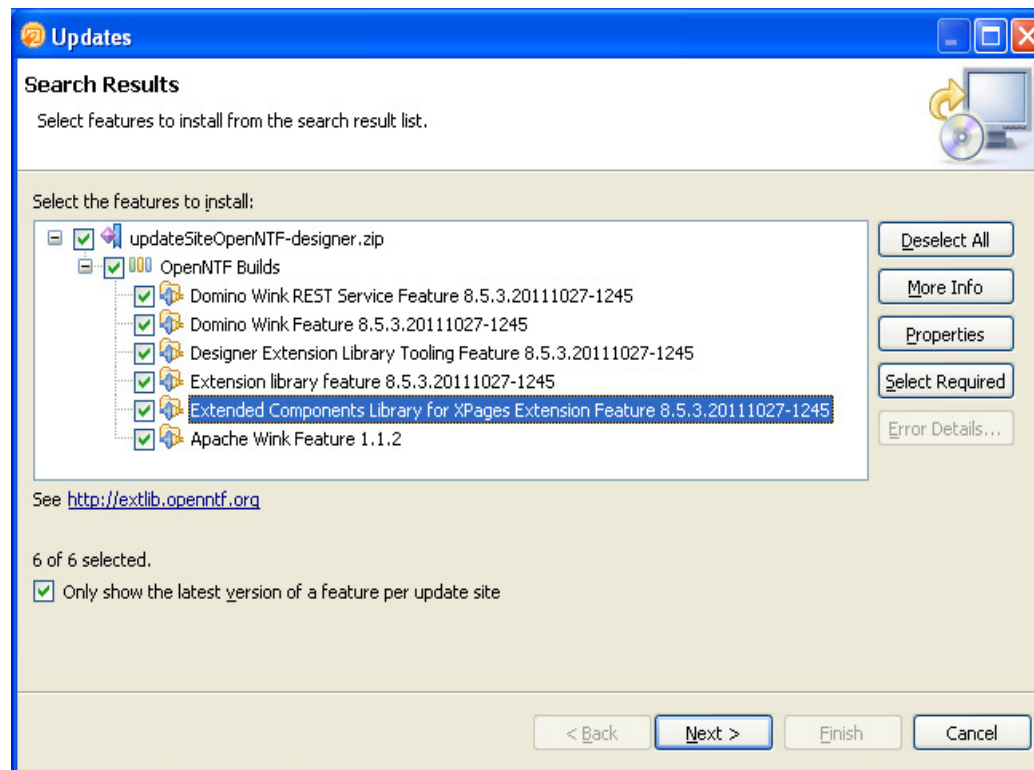


- You need to point Domino Designer to the location of updateSiteOpenNTF-designer.zip which was previously extracted from the big OpenNTF zip file



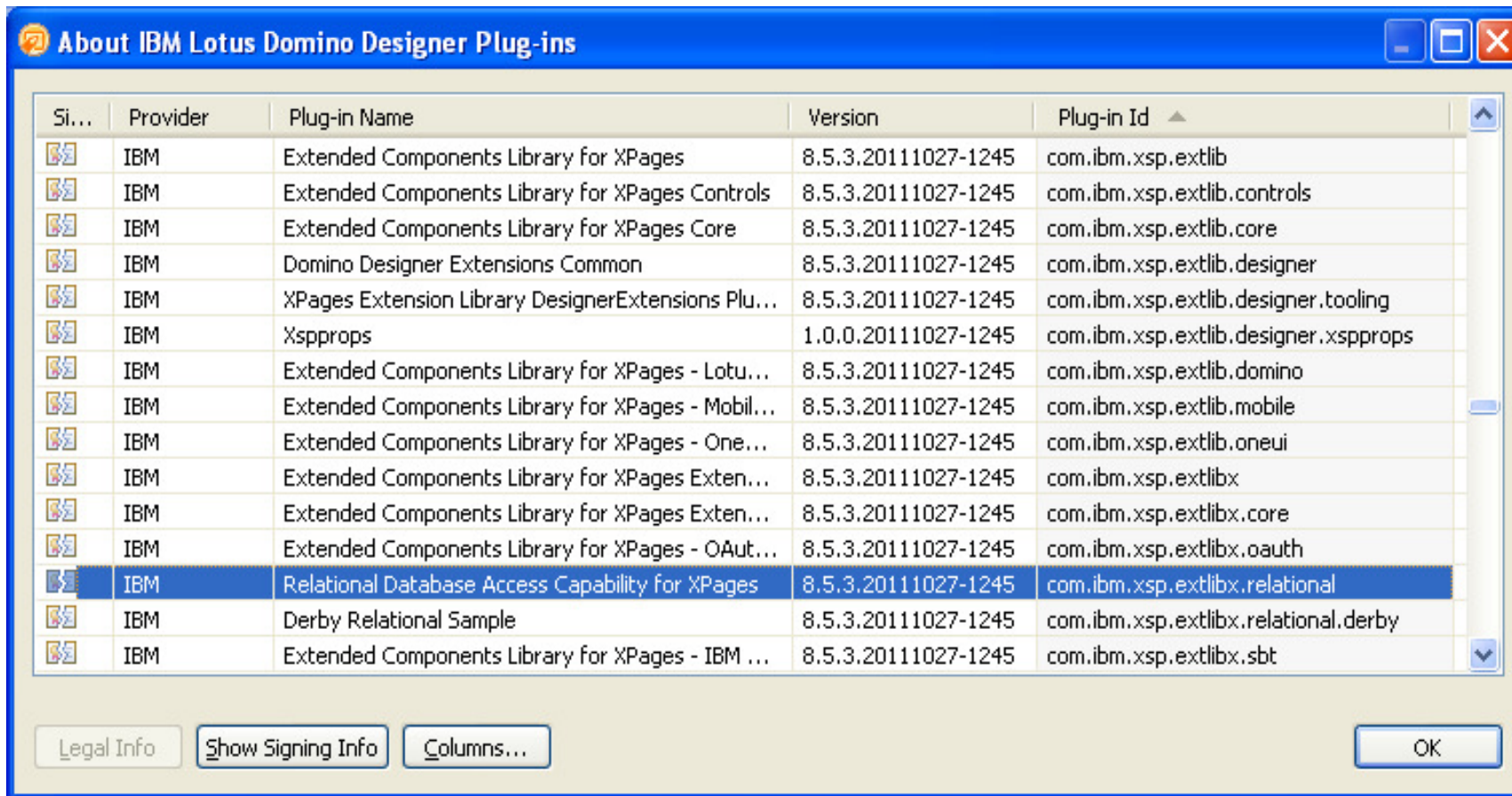
Setup of server/development environments

- The Domino Designer update site installation process displays the features which are available in selected update site
- One of these features is related to extlibx library functionality, which contains Relational support
- During further steps you'll need to confirm, several times, that you'd like to install the unsigned features



Setup of server/development environments

- Verification of the correct installation within Domino Designer:
 - Select Help → About IBM Lotus Domino Designer menu
 - Click Plug-in Details button, sort by Plug-in Id and scroll to the extlib/extlibx plugins



Future Directions for RDBMS Support

- XPages Extension Library OpenNTF and Relational support continues to evolve further and expected features in the pipeline include:
 - Global Domino server based RDBMS connection definition ability
 - Better Domino Designer tooling to work with the RDBMS tables' metadata
 - Even better JDBC connection pooling implementation using Apache DBCP
 - JDBC Query possibility via REST services
 - Additional @Functions you are missing for your next great XPages application ;-)

Agenda

- Access to relational Databases from XPages
- JDBC Driver to access NSF's
- Domino REST APIs
- Workflow for XPages
- OSGi Tasklet Service

Domino JDBC Access

- Currently a preview technology delivered on openNTF
 - <http://bit.ly/xn9PF1>
- Expose Domino data to external applications
 - Allow data analytics on top of Domino data
 - Fully support the Notes/Domino security model – Use the Internet user/password
 - Connect from any application that supports JDBC connectivity
- Augment the Domino query capability with a new SQL engine working on top of the Domino data (views)
 - Use SQL for dynamic queries, including custom selection, aggregation operators...
 - Supports very complex queries on top of Domino data
 - Dynamically re-sort view entries without creating an index
 - Join data between multiple views, even coming from different databases/servers
 - Aggregate data from different views/databases/servers
 - Bring all the power of SQL to Domino data :-)
 - Make it easy to consume from an XPages application
 - Tightly integrated with the XPages Extension Library RDBMS extension
 - Execute queries with the rights of the XPages user (a.k.a “Web User”)

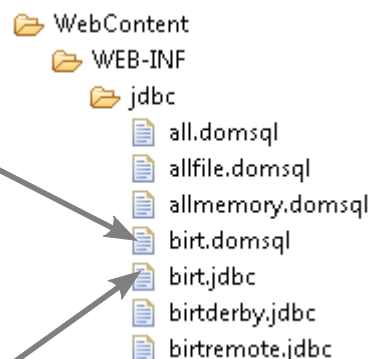
DomSQL in a Nutshell

- Gives read-only access to the data, by projecting the content of views as relational tables
- Exposes the Domino data through a JDBC interface
 - Includes data and metadata access
- DomSQL is based on top of an award winning SQL engine: SQLite
 - Very robust SQL engine used by many projects (<http://www.sqlite.org/>)
 - Well known SQL grammar understood by many JDBC clients
- Multiple execution modes
 - Executes within the HTTP process for XPages consumption
 - Maximum performance, no network connection involved
 - Executes within the DOTS process when accessed remotely
 - Remote server implemented using RMI
- The engine is fully extensible, if needed
 - The source code is provided under the Apache license, and it is only using the “C” toolkit public API.

Use it From XPages with the Extension Library

**Relational data
definition**

**JDBC Connection
(Extension Library)**



**Relational Datasource
(Extension Library)**

Show data from: JDBC Query

Property	Value
calculateCount	
clearOnRendering	
connectionManager	jdbcConnectionManager1
connectionName	
connectionUrl	
defaultOrderBy	

Connection: Birt

Select a Table: EMPLOYEES

**Any XPages Control that can bind to data
(views...)**

_ID	EMPLOYEEENUMBER	LASTNAME	FIRSTNAME	EXTENSION	EMAIL
765058	1002	Murphy	Diane	x5800	dmurphy@classicmode
765062	1056	Patterson	Mary	x4611	mpatterson@classicmoc
765066	1076	Firrelli	Jeff	x9273	jfirrelli@classicmodelca
765070	1088	Patterson	William	x4871	wpatterson@classicmo

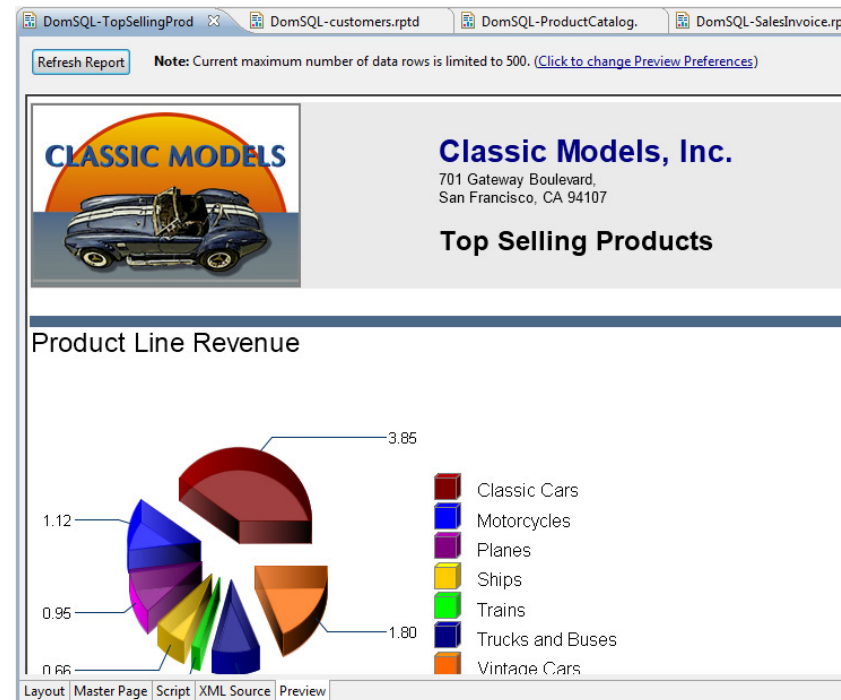
Or From Any Third Party Application That Supports JDBC Connectivity

The screenshot shows the DBVizualizer application interface. On the left, a tree view under 'Connections' shows a 'DomSQL Remote Server' with a 'TABLE' folder containing various tables, with 'CUSTOMERS' selected. The main pane shows 'Table: CUSTOMERS' with a toolbar for 'Info', 'Columns', 'Data', 'Row Count', 'Primary Key', and 'Indexes'. The 'Data' tab is active, displaying a table with 7 rows and 4 columns: _ID, CUSTOMERNUMBER, CUSTOMERNAME, and a fourth column with single letters. The data is as follows:

_ID	CUSTOMERNUMBER	CUSTOMERNAME	
1	764570	103.0 Atelier graphique	S
2	764574	112.0 Signal Gift Stores	K
3	764578	114.0 Australian Collectors, Co.	F
4	764582	119.0 La Rochelle Gifts	L
5	764586	121.0 Baane Mini Imports	B
6	764590	124.0 Mini Gifts Distributors Ltd.	N
7	764594	125.0 Havel & Zbyszek Co	P

DBVizualizer

Birt Report Editor



Agenda

- Access to relational Databases from XPages
- JDBC Driver to access NSFs
- Domino REST APIs
- Workflow for XPages
- OSGi Tasklet Service



Types of REST Services

	Component Type	Container	Data Model	Data Format	Deployment
Domino Data Service	OSGi plug-in	OSGi framework	Databases, views, documents, etc.	JSON	Installed with extension library
XPages REST Services Control	XPages custom control	XPage	Databases, views, documents, etc.	JSON or XML	You add a control to an XPage
Custom Database Servlet	Java class	Database	Databases, views, documents, etc.	JSON or XML	You add a Java class to a database design
Custom Wink Servlet	OSGi plug-in	OSGi framework	Good for a higher level of abstraction (e.g. calendar objects)	Any format you choose	You create the OSGi plug-in and add it to the Domino OSGi framework.



Domino Data Service Resources

- Database collection – list of databases on a server
 - <http://{host}/api/data>
- View/folder collection – list of views and folders in a database
 - <http://{host}/{database}/api/data/collections>
- View entry collection – list of entries in a view (or folder)
 - <http://{host}/{database}/api/data/collections/unid/{unid}>
 - <http://{host}/{database}/api/data/collections/name/{name or alias}>
- View entry – for updating a view (or folder) entry
 - <http://{host}/{database}/api/data/collections/unid/{unid}/unid/{unid}>
 - <http://{host}/{database}/api/data/collections/name/{name or alias}/unid/{unid}>
- Document collection – for creating new documents
 - <http://{host}/{database}/api/data/documents>
- Document – for reading, updating and deleting existing documents
 - <http://{host}/{database}/api/data/documents/unid/{unid}>



Enabling the data service is required

- Data service is disabled by default.
- Can be enabled for Server, Database (View and Document), and View

```
{  
  "code":403,  
  "text":"Forbidden",  
  "message":"Database not allowed for Web Access"  
}
```

- **Administrator** controls which servers run the data service:

Domino Access Services

The following setting is a place holder for services provided by an external plug-in. See the Lotus Notes and Domino wiki for more information.

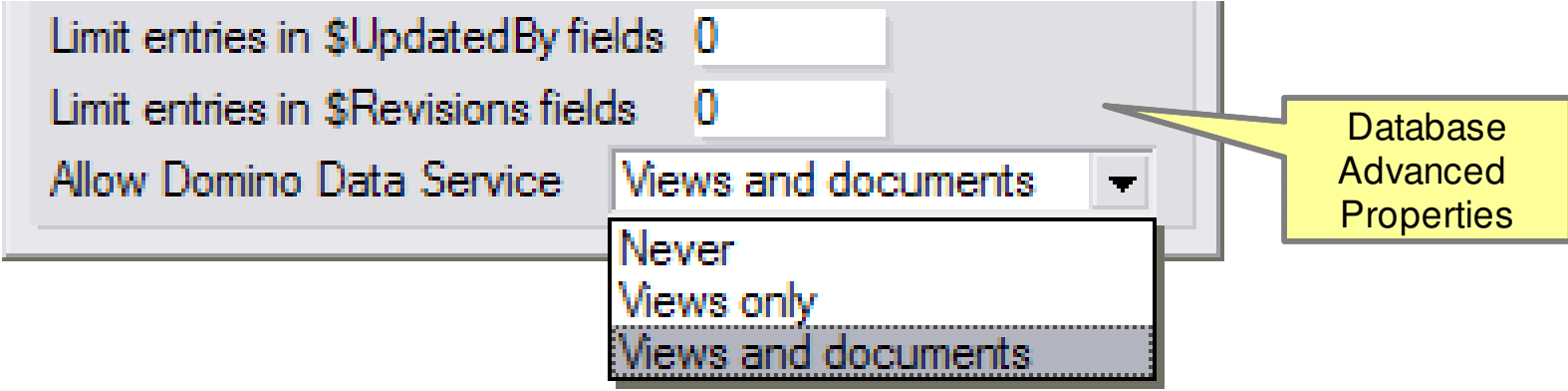
Enabled services: 『Data』 ▼

Internet Site
or Server
Document



Enabling the data service for a database

- Application developer controls **Database** access:



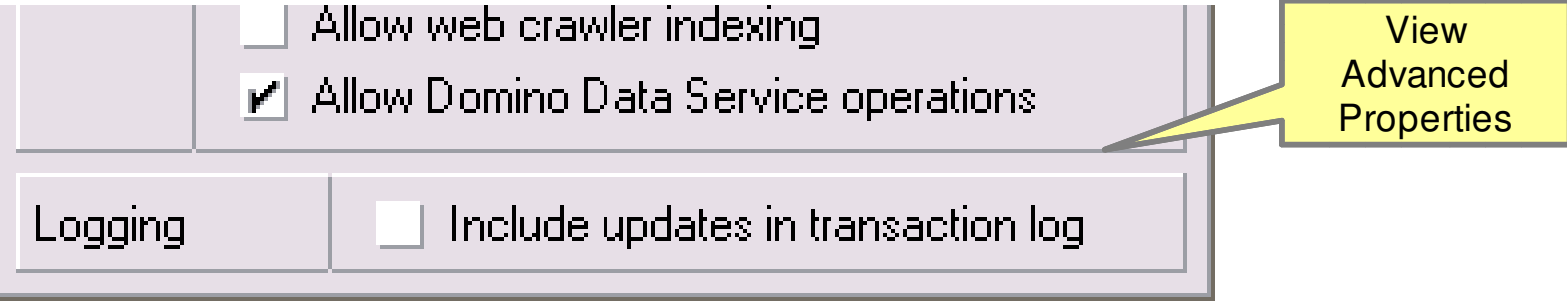
The screenshot shows the 'Database Advanced Properties' dialog box. It contains three input fields: 'Limit entries in \$UpdatedBy fields' with the value '0', 'Limit entries in \$Revisions fields' with the value '0', and 'Allow Domino Data Service' with a dropdown menu. The dropdown menu is open, showing three options: 'Never', 'Views only', and 'Views and documents'. A yellow callout box points to the dialog box with the text 'Database Advanced Properties'.

Limit entries in \$UpdatedBy fields	0
Limit entries in \$Revisions fields	0
Allow Domino Data Service	Views and documents

Never
Views only
Views and documents

Database Advanced Properties

- Application developer controls **View** access:



The screenshot shows the 'View Advanced Properties' dialog box. It contains two checkboxes: 'Allow web crawler indexing' (unchecked) and 'Allow Domino Data Service operations' (checked). Below these is a section labeled 'Logging' with a checkbox for 'Include updates in transaction log' (unchecked). A yellow callout box points to the dialog box with the text 'View Advanced Properties'.

<input type="checkbox"/> Allow web crawler indexing
<input checked="" type="checkbox"/> Allow Domino Data Service operations
Logging
<input type="checkbox"/> Include updates in transaction log

View Advanced Properties



RESTful Domino

- Domino REST Services can be accessed two ways
 - From XPages using the REST Service control
 - As built-in service called Domino Access Services (DAS)
- Domino Access Services (DAS) is an extensible set of services
 - Domino Data Service
 - Mail Service (Planned)
 - Calendar Service (Planned)
 - Custom Services

Agenda

- Access to relational Databases from XPages
- JDBC Driver to access NSFs
- Domino REST APIs
- Workflow for XPages
- OSGi Tasklet Service

Objectives

- Promote a common Workflow capability across products
 - Make it platform agnostic with drivers that encapsulate the platform differences
 - Full based on top of Java/OSGi
 - Come with a robust implementation dedicated to the Domino platform
 - Allow the move to a more robust architecture, like Lombardi
- Increase the Domino Application Development capability by providing an easy to use Workflow capability
 - Increase the value of Domino as an application development platform
 - Better compete with Microsoft Sharepoint, and .NET
 - Workflow is one of the Domino app dev selling point, so make it real!

Design Goals

- Make the workflow engine easy to use and integrate for the simple cases
 - Predefined Workflow templates
 - Simple approval...
 - Easy to parameterize
 - Connects to the existing infrastructure (directory, ...)
 - Built-in drivers
 - Ready to go set of XPages controls/API
- Make the engine customizable for more complex cases
 - Custom workflow templates
 - Requires some custom Java/XPages custom development
- Developer can use similar API to access workflow data regardless of the workflow type
 - A workflow context object exposes the current state of the workflow and allows actions to be triggered
- Ensure that nothing is “hard coded” but provided through pluggable services
 - From an XPages standpoint, the services will be available through Complex types

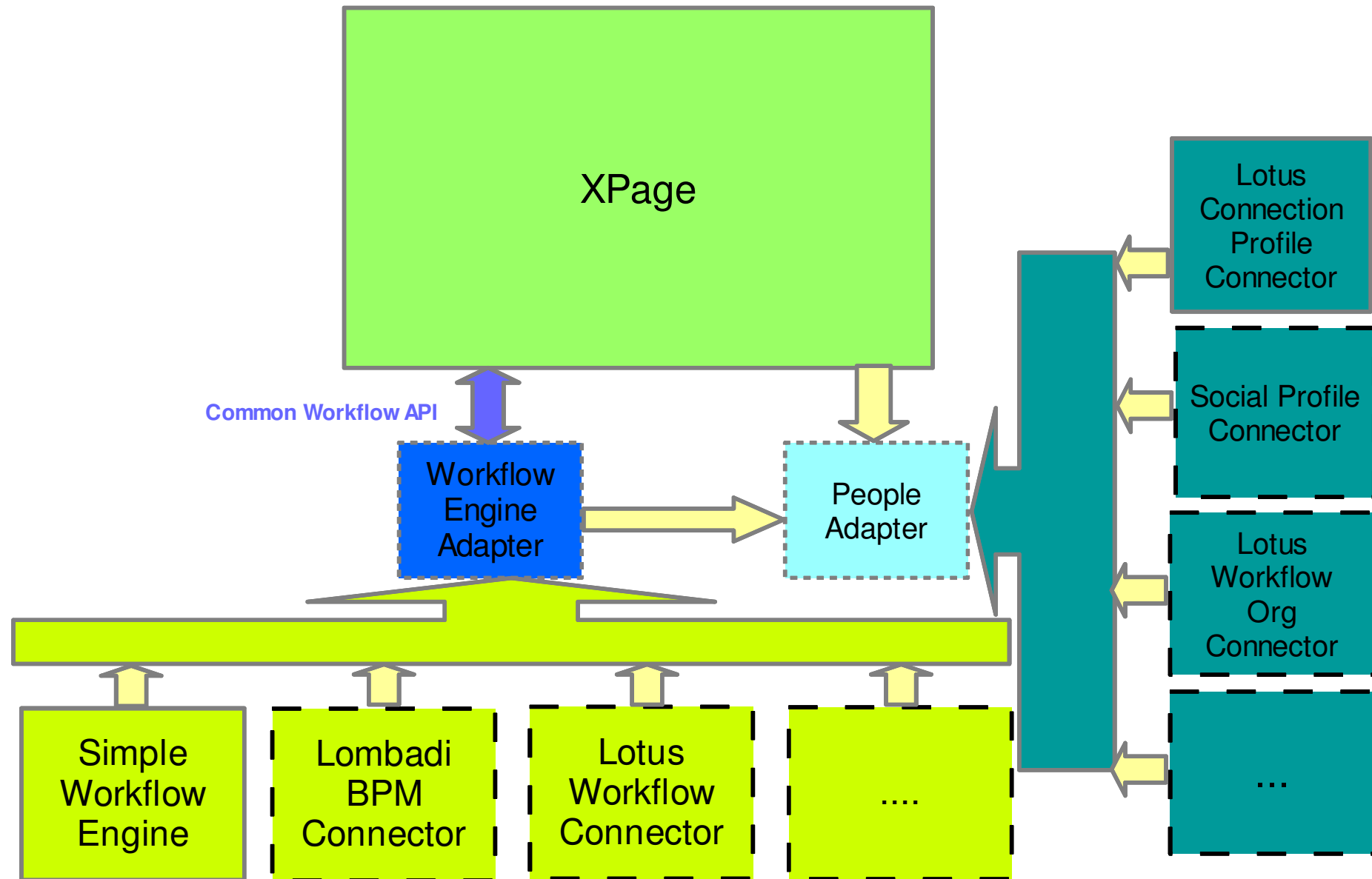
Design Goals

- Make the workflow engine easy to use and integrate for the simple cases
 - Predefined Workflow templates
 - Simple approval...
 - Easy to parameterize
 - Connects to the existing infrastructure (directory, ...)
 - Built-in drivers
 - Ready to go set of XPages controls/API
- Make the engine customizable for more complex cases
 - Custom workflow templates
 - Requires some custom Java/XPages custom development
- Developer can use similar API to access workflow data regardless of the workflow type
 - A workflow context object exposes the current state of the workflow and allows actions to be triggered
- Ensure that nothing is “hard coded” but provided through pluggable services
 - From an XPages standpoint, the services will be available through Complex types

Workflow types

- Different types of workflows are available
 - Ad hoc workflows, for simple, no administration workflows
 - Defined as part of the forms, using simple components
 - Similar to the Workflow sub form
 - Changing a process means opening Domino Designer and editing an XPage component
 - Actions are happening synchronously
 - Managed Workflows
 - Defined outside of the form, generally is external databases
 - Actions are happening asynchronously, using a background router
- Workflow context abstraction
 - The workflow is accessed the same way, regardless of the underlying implementation
- XPages common controls
 - Comes with a set of ready to use controls (action bar, display history...), connected to the workflow context

Architecture



Demo app snapshot: manager list all employee's "End of Year Review"

XPAGES Home iPhone Rendering Qian Liang Help

End Of Year Review [Setup](#)










My Reviews

My Employees

Pending Tasks

Create Review

[Collapse All](#) | [Expand All](#) Previous 1 Next

	State
 2012: test4 liang	Objectives submitted - Waiting for approval
 2012: test1 liang	Assessment submitted - Waiting for evaluation
 2012: Mike Motler	Assessment to be submitted by the employee
 2012: Ron Espinosa	Objectives to be submitted
 2012: Simone Dray	Objectives submitted - Waiting for approval
 2012: Vijay Nehry	Objectives submitted - Waiting for approval
 2012: Kelly Hardart	Assessment to be submitted by the employee
 2012: Lucille Suarez	Evaluation Approved
 2012: Ted Amado	Objectives submitted - Waiting for approval

Show 10 | **25** | 50 | All items per page



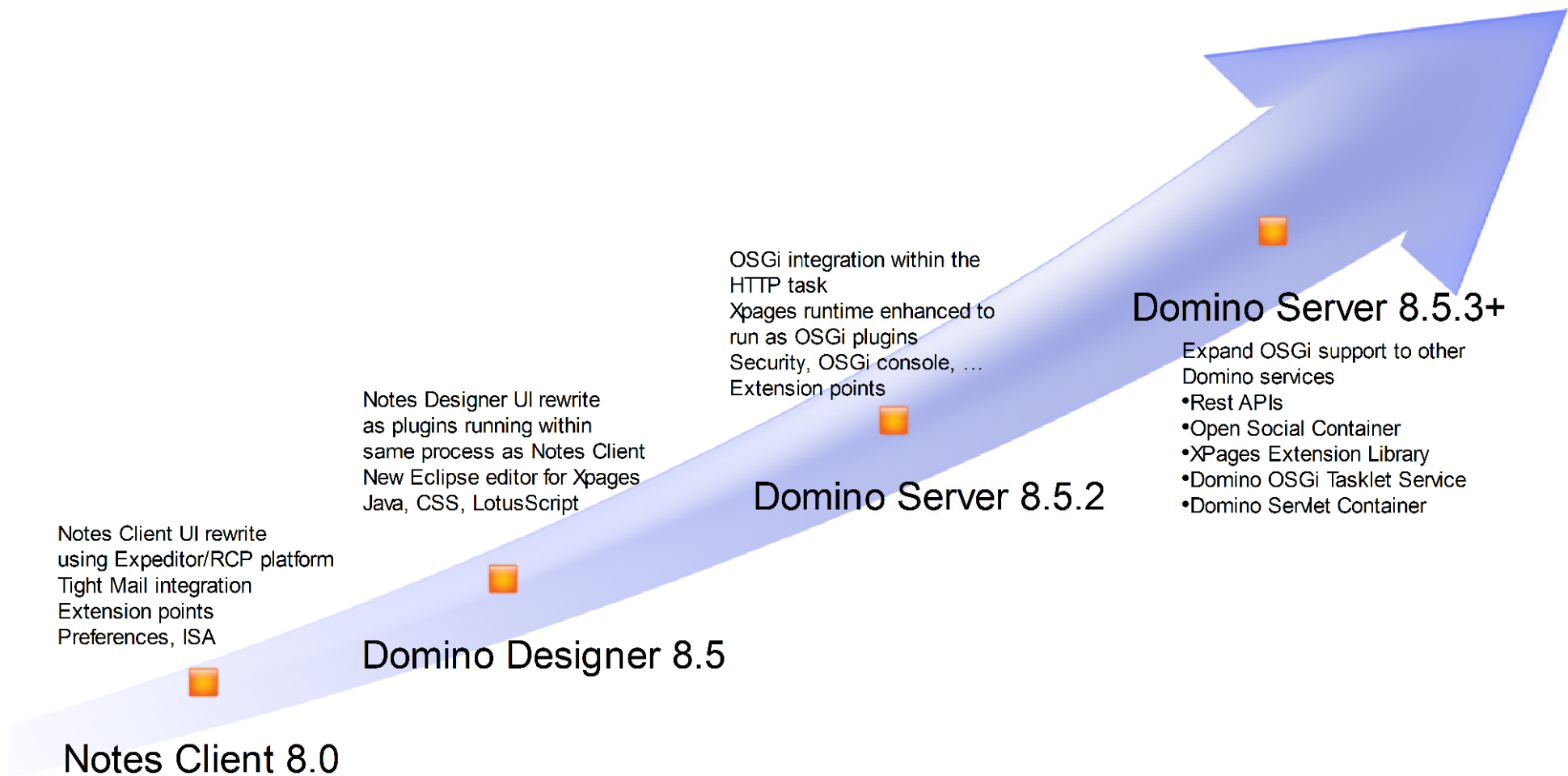
Agenda

- Access to relational Databases from XPages
- JDBC Driver to access NSFs
- Domino REST APIs
- Workflow for XPages
- OSGi Tasklet Service
 - Introduction and technology overview
 - Environment setup
 - Creating, deploying & debugging tasklets

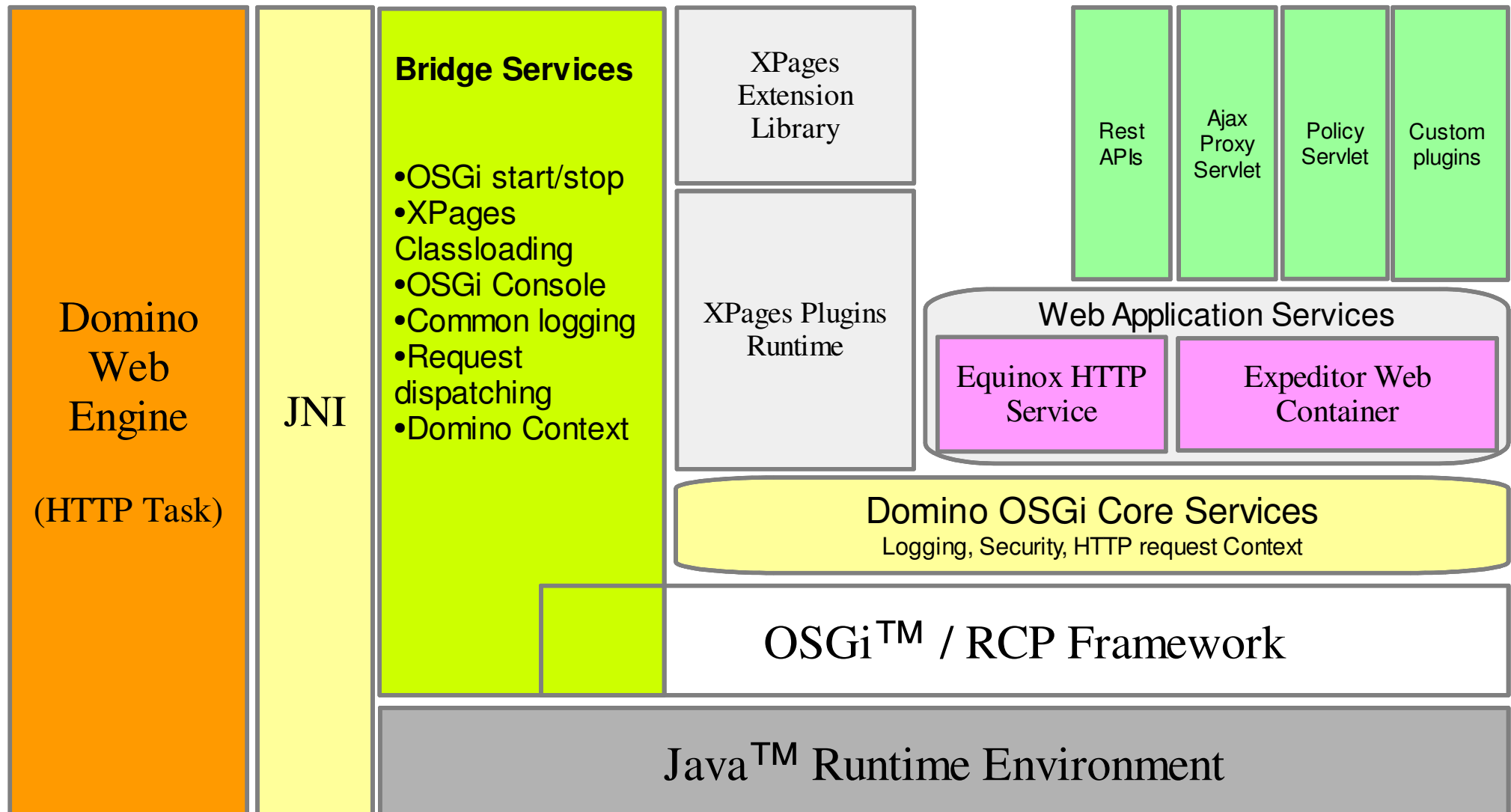
Introduction to OSGi™

- OSGi used to stand for Open Service Gateway Initiative but is now just a trademark used to define the specifications for a module system and service platform
- Component model: OSGi bundle
 - Private classpath: each bundle has its own classloader
 - Static resources
 - Fine grained dependencies control via package import/export
- Many other capabilities available
 - Easy deployment
 - Hot installation and dynamic configuration
 - Provisioning
 - Lifecycle via Bundle activator
- Multiple open source implementations
 - Eclipse equinox
 - Apache felix, Knopflerfish, Karaf, Concierge...

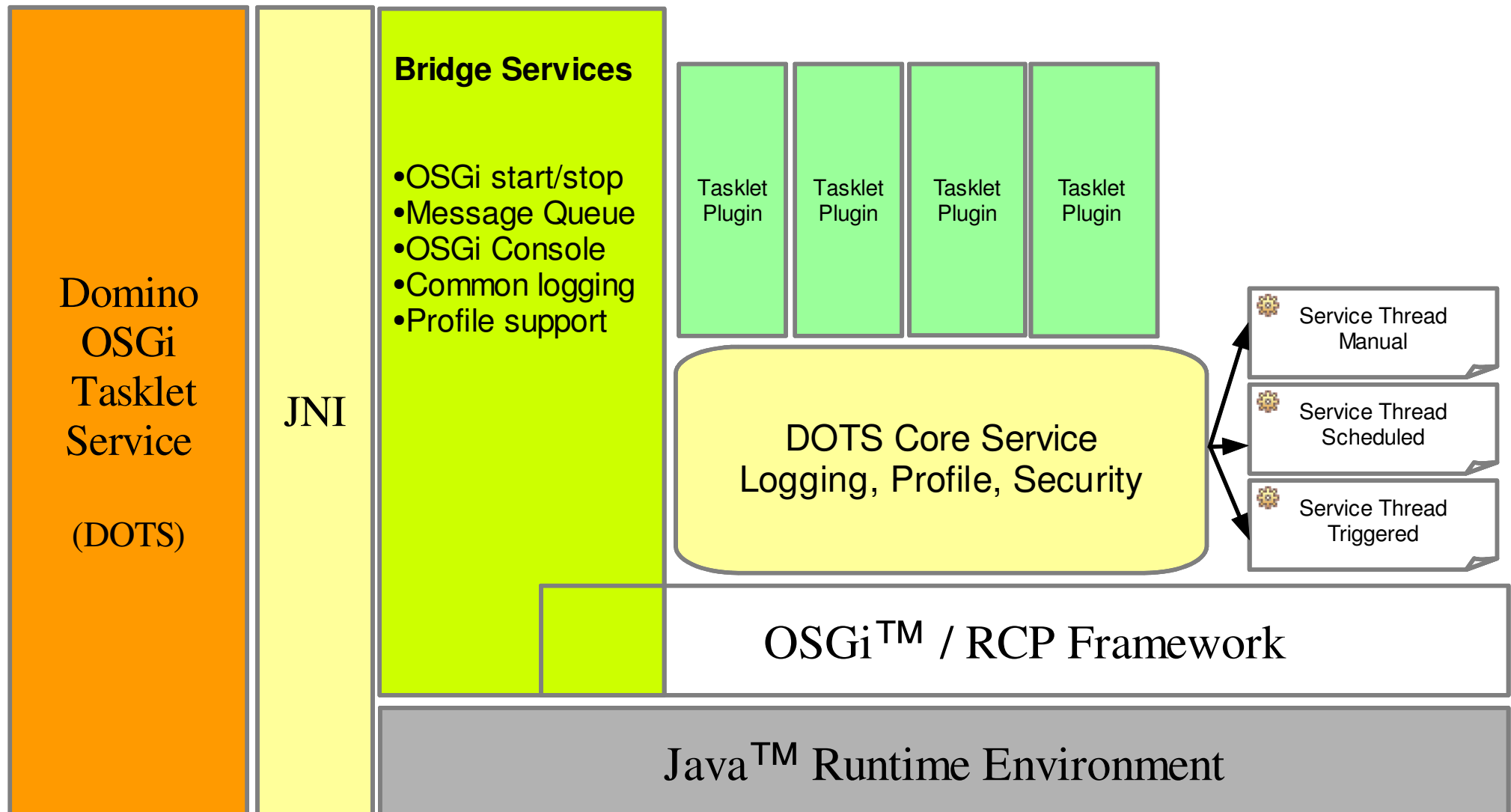
Where is it used in IBM Lotus® Notes®/Domino®?



IBM Lotus Domino OSGi in HTTP architecture



IBM Lotus Domino OSGi Tasklet Service architecture



Multiple deployments

- File based
 - Bundles are packaged as jars or directories and dropped into the server file system
 - Recommended location is {DominoData}/domino/workspace/applications
 - Bundles execute with maximum security privileges
- NSF Based
 - Bundles are imported into a database using the UpdateSite.ntf template
 - Server is configured to use the updateSite database (notes.ini variable for HTTP, Profile for DOTS)
 - Documents containing the bundles must be signed with privileged user id to load, after which bundles execute with maximum security privileges
- PDE tool: Developer configuration
 - Server is instrumented to run bundles directly from eclipse workspace: deployment-less
 - No need to export bundles every time a change is made
 - Bundle are not physically installed making it easy to switch between configurations

Prerequisites

- IBM Lotus Domino Server, 8.5.2+
 - <http://www.ibm.com/developerworks/downloads/ls/lsts>
- Eclipse, 3.6.2+
 - <http://www.eclipse.org/downloads>
- IBM Lotus Domino Debug Plugin (PDE tool)
 - <http://www.openntf.org/internal/home.nsf/release.xsp?documentId=CBF874E9C4607B4C8625799D00287B8C>
- OSGi Tasklet Service for IBM Lotus Domino, 2.0.3+
 - <http://www.openntf.org/internal/home.nsf/release.xsp?databaseName=CN=NotesOSS2/O=NotesOSS!!Projects/pmt.nsf&documentId=84018B5D35E44E2F8625795E00140D3A&action=openDocument>
- XPages Extension Library
 - Domino 8.5.3 Upgrade Pack 1
 - or <http://www.openntf.org/internal/home.nsf/release.xsp?databaseName=CN=NotesOSS2/O=NotesOSS!!Projects/pmt.nsf&documentId=2E25B5DA1BDFB9B48625794D005B0821&action=openDocument>

Environment setup for DOTS

- Installing DOTS
 - Required Eclipse plug-ins and DOTS core files
- PDE Tool
 - The version installed during the HTTP environment setup will persist across workspaces so there is no action needed. On a clean setup you would follow the steps previous demonstrated in the HTTP environment setup.
- Target Platform
 - DOTS is just another OSGi setup so it's almost the same as the HTTP setup with a few minor differences.
- Notes.jar
 - This plug-in is just used for development and will be exactly the same as the version in the HTTP environment. Simply import from the other workspace. On a clean setup you would follow the steps previous demonstrated in the HTTP environment setup.

Installing DOTS (Required Eclipse plug-ins)

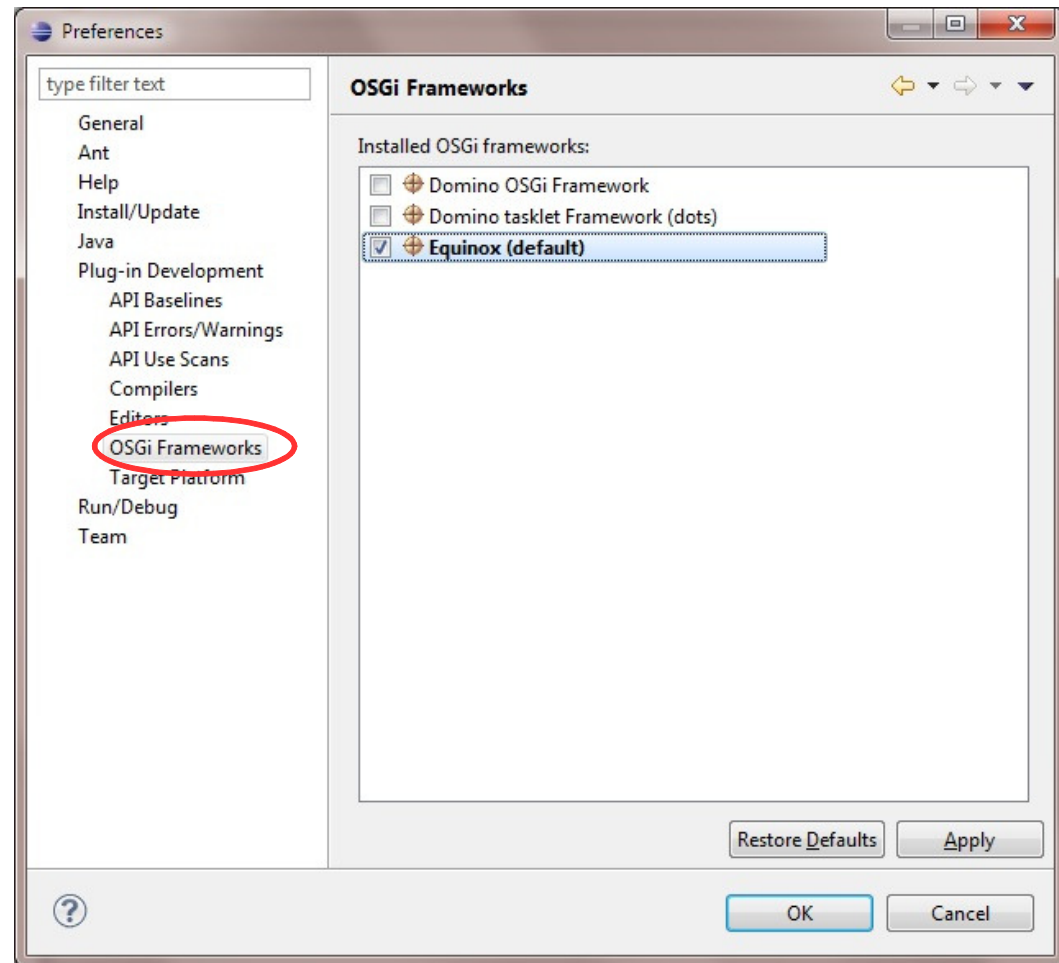
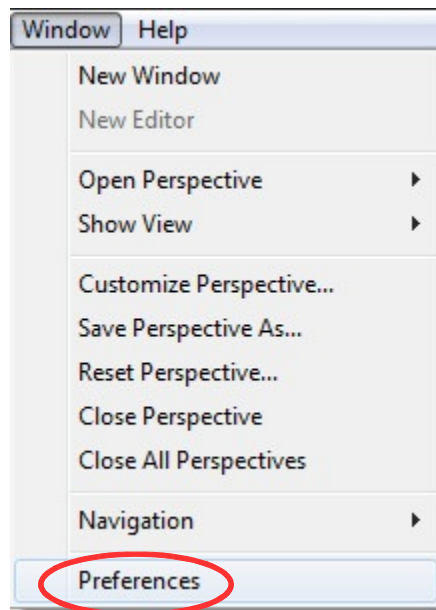
- The OSGi framework is located in {DominoBin}/osgi-dots/rcp/eclipse
- The minimum required set of plug-ins is comprised of the following list:
 - org.eclipse.core.contenttype
 - org.eclipse.core.jobs
 - org.eclipse.core.runtime.compatibility.auth
 - org.eclipse.core.runtime
 - org.eclipse.equinox.app
 - org.eclipse.equinox.common
 - org.eclipse.equinox.preferences
 - org.eclipse.equinox.registry
 - org.eclipse.osgi
 - org.eclipse.update.configurator
- You should be able to simply copy these 10 plug-ins into the {DominoBin}/osgi-dots/rcp/eclipse/plugins directory from any installed version of the Eclipse SDK.

Installing DOTS (DOTS core files)

- Download the OSGI Tasklet Service for IBM Lotus Domino from OpenNTF
- The following files must be installed in the {DominoBin} directory
 - ndots.exe (Windows)
 - dots, dots.res (UNIX)
- The following files must be installed in the {DominoBin}/osgi-dots directory
 - launcher.jar
 - dotssec.jar
- The following files must be installed in the {DominoBin}/osgi-dots/shared/eclipse/plugins directory
 - com.ibm.dots_2.0.3.XXXX.jar
 - com.ibm.dots.samples_2.0.3.XXXX.jar (optional)

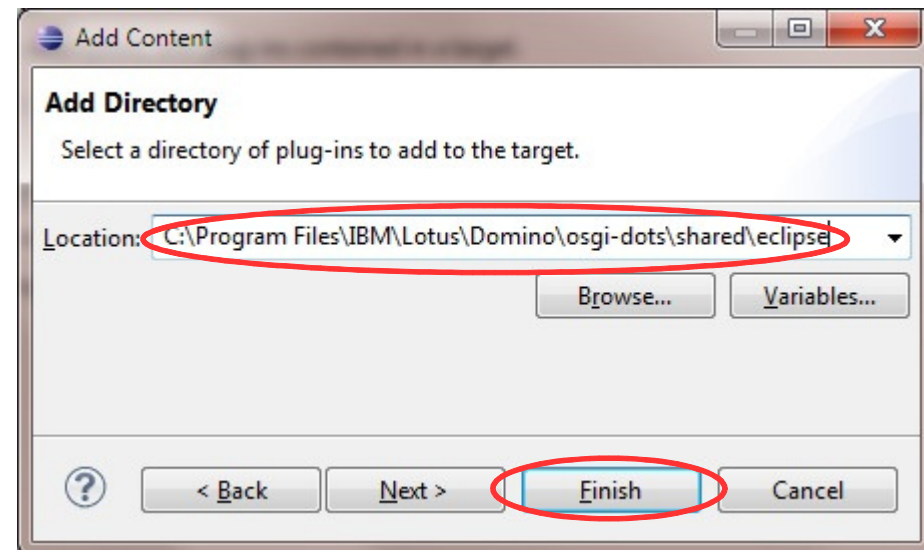
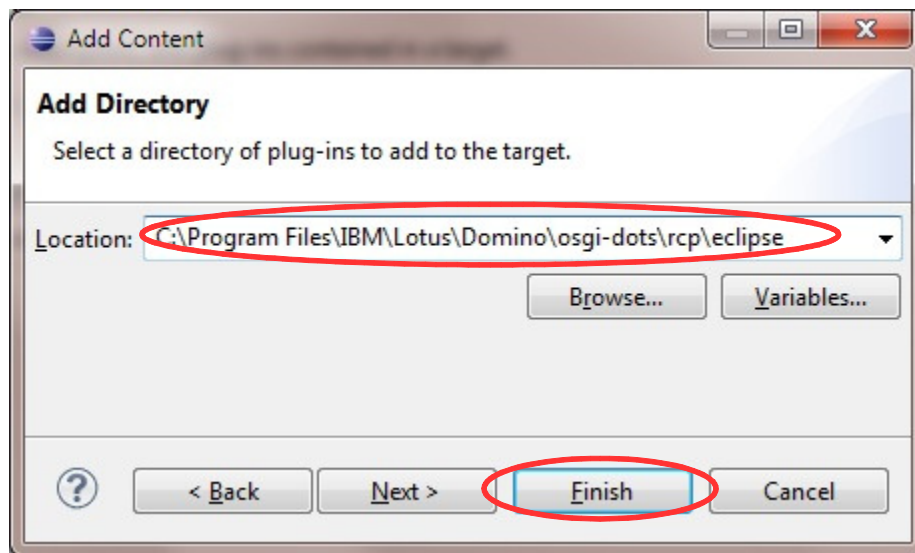
Environment setup for DOTS (PDE tool recap)

- An easy verification that the PDE Tool is installed is to check the OSGi Frameworks available to Eclipse
- You should see 3 frameworks
 - Domino OSGi Framework
 - Domino tasklet Framework (dots)
 - Equinox



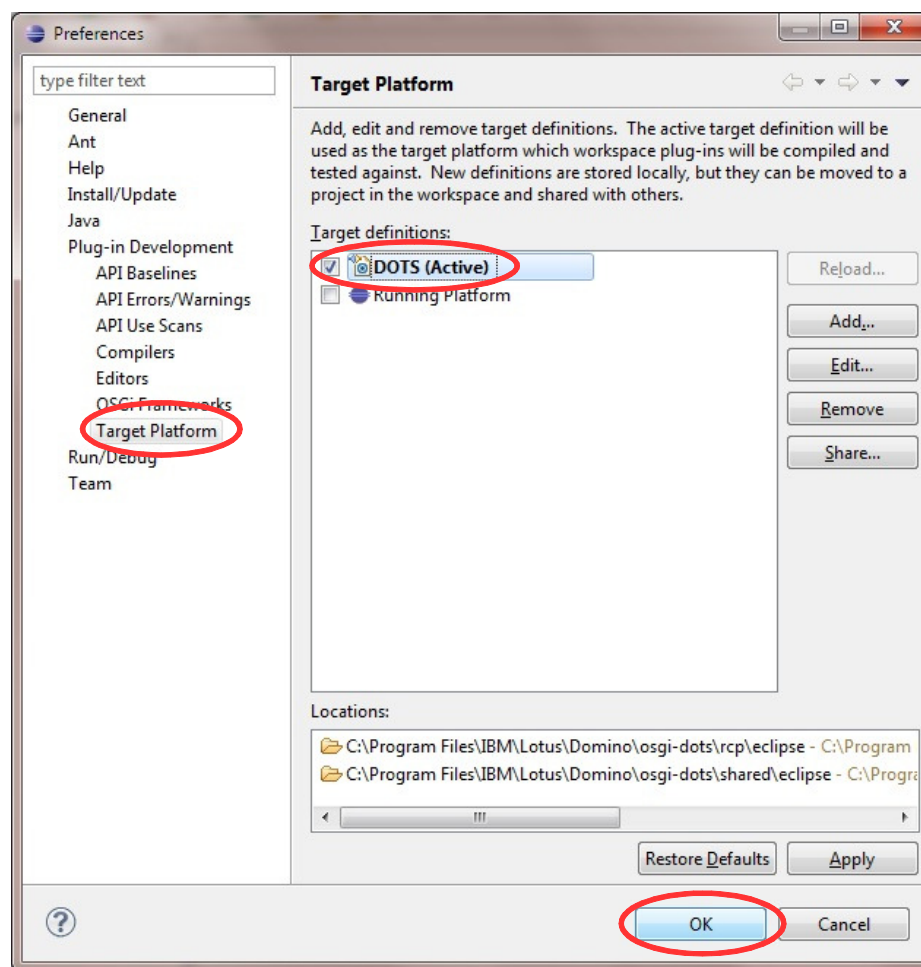
Environment setup for DOTS (Target Platform recap)

- Add the {DominoBin}\osgi-dots\rcp\eclipse directory and click the Finish button
- Repeat the process to add the {DominoBin}\osgi-dots\shared\eclipse directory



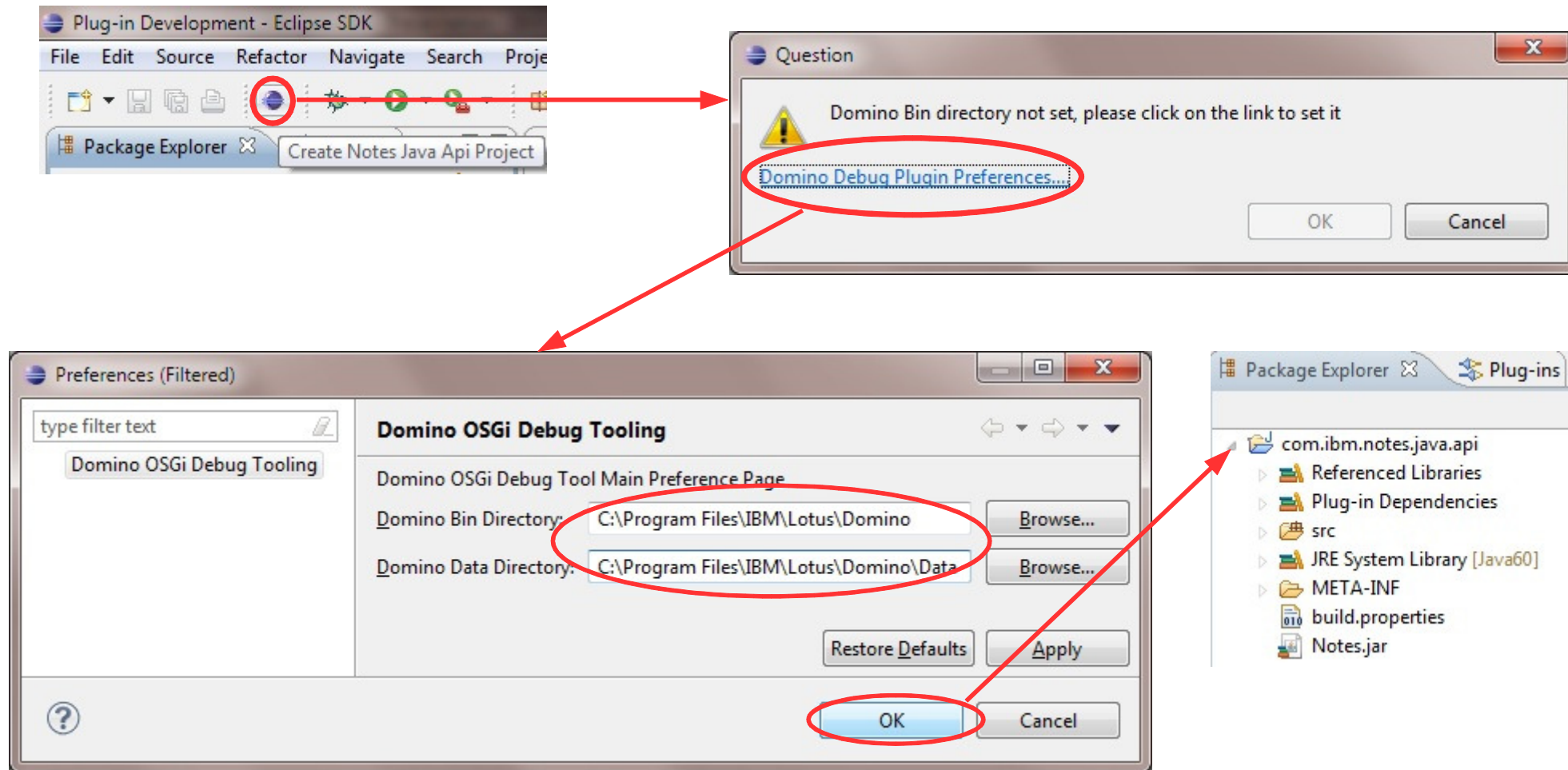
Environment setup for DOTS (Target Platform recap)

- Select the new Target definition you just created and click the OK button

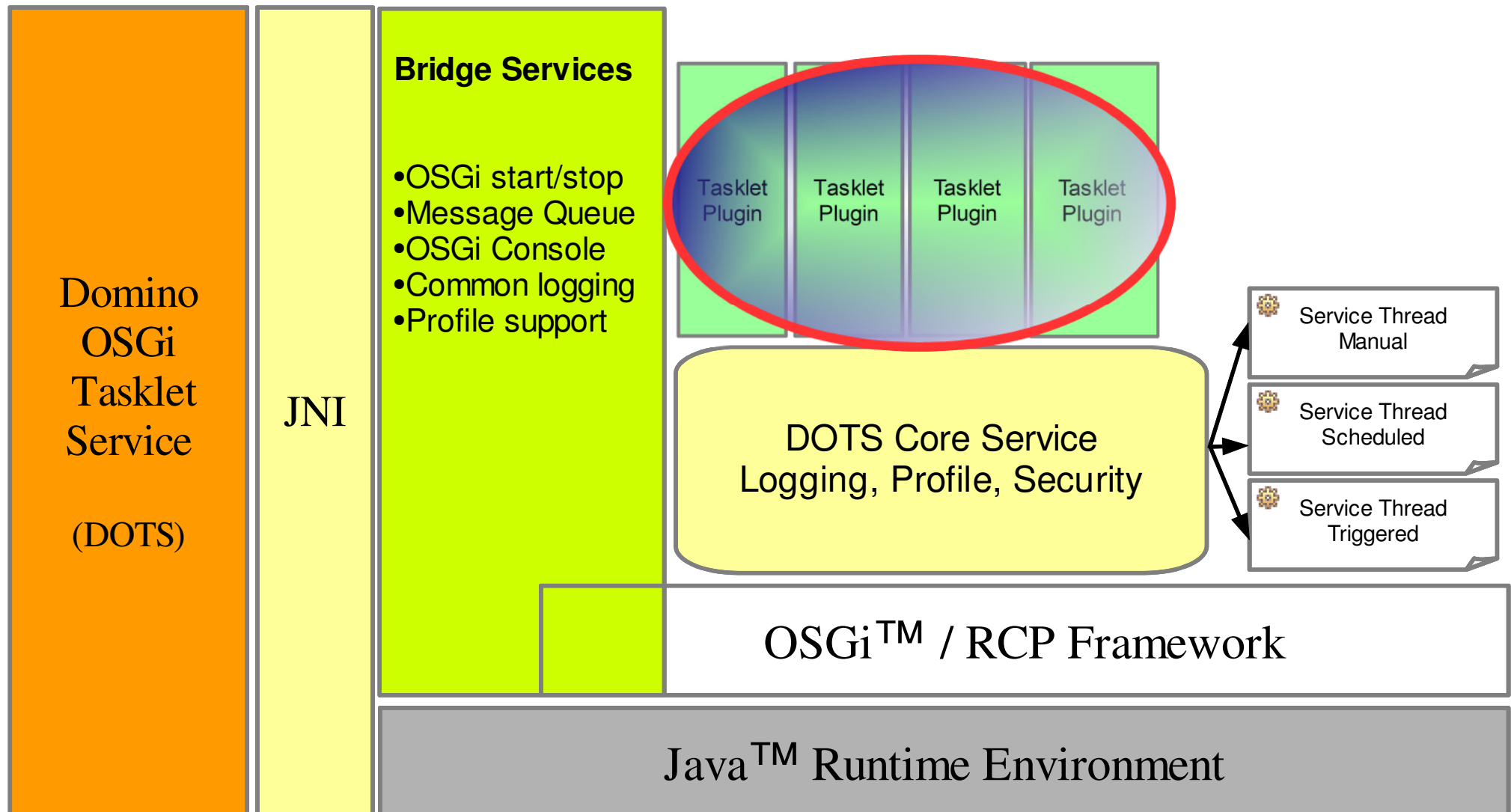


Environment setup for DOTS (Notes.jar recap)

- Use the “Create Notes Java Api Project” option from the PDE tool

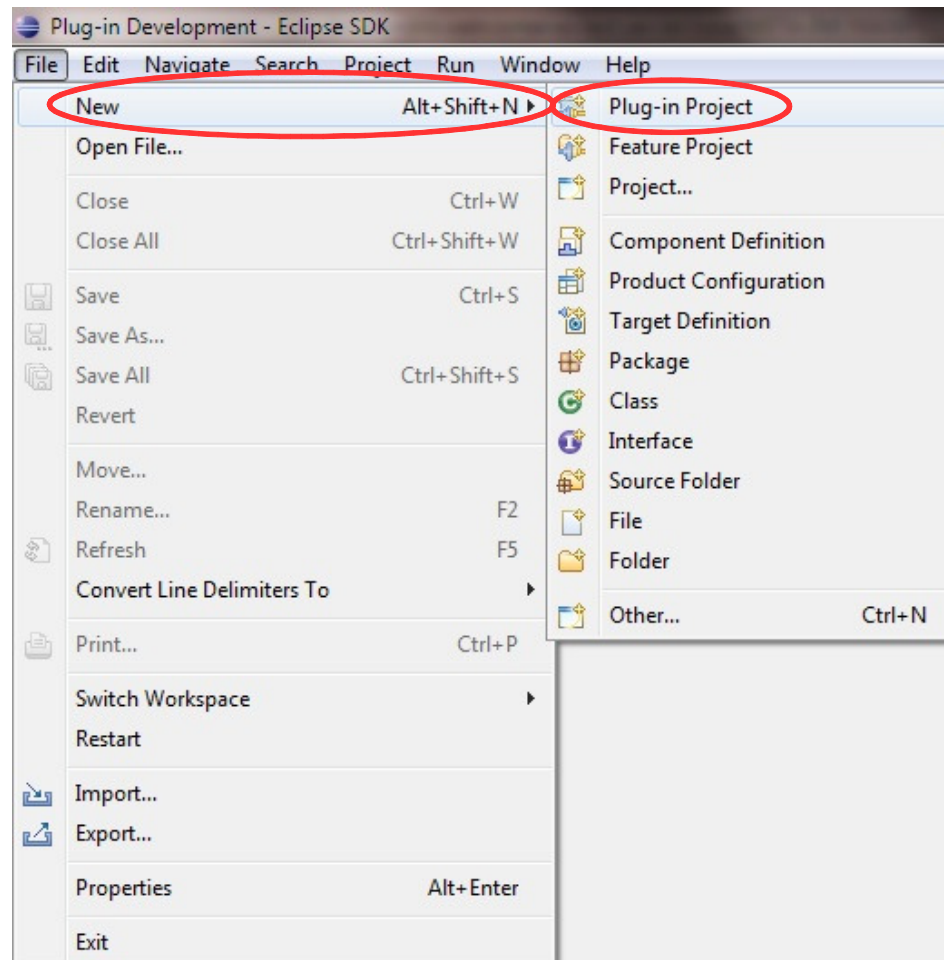


Creating DOTS tasklets



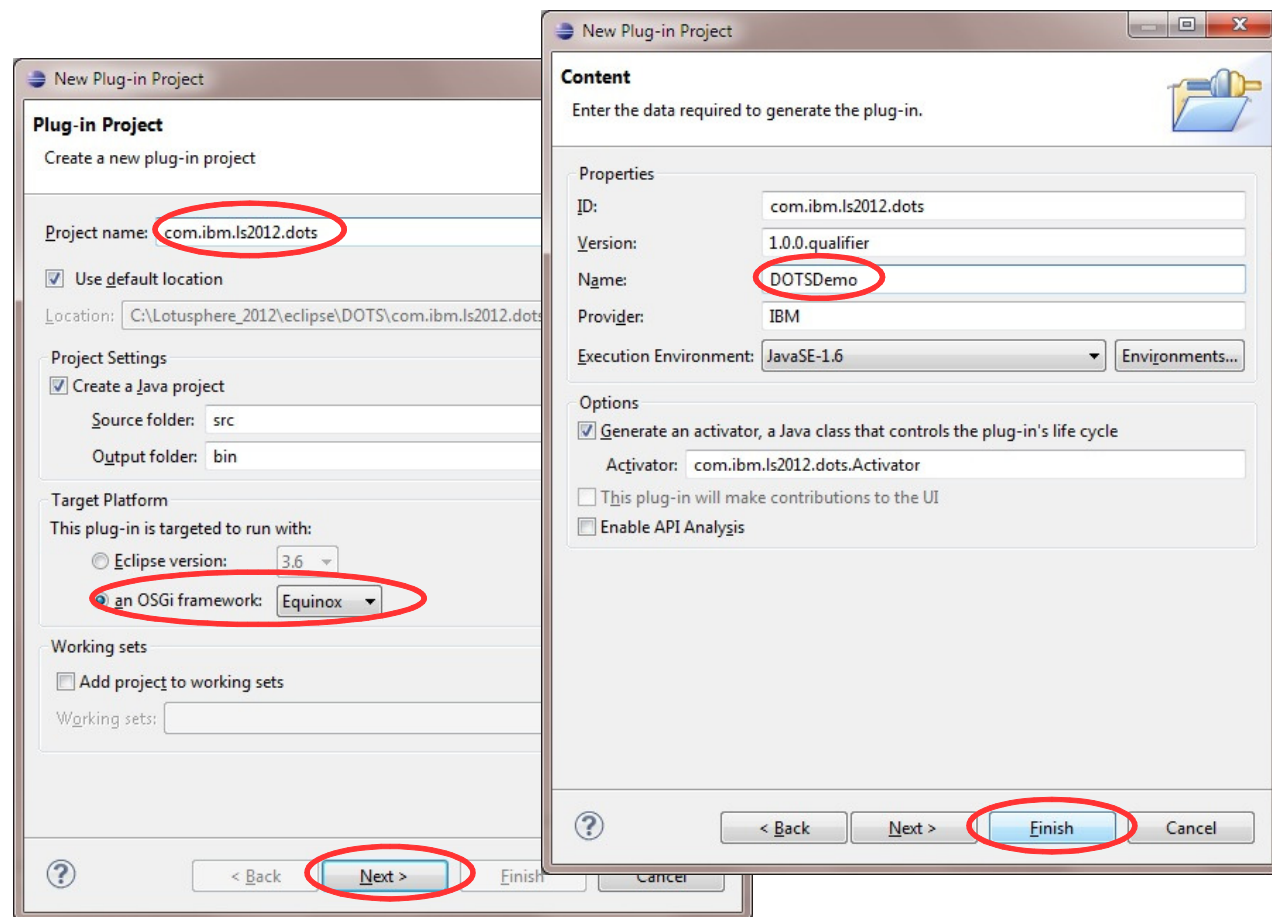
Creating DOTS tasklets

- Create a new workspace plug-in
 - File->New->Plug-in-Project



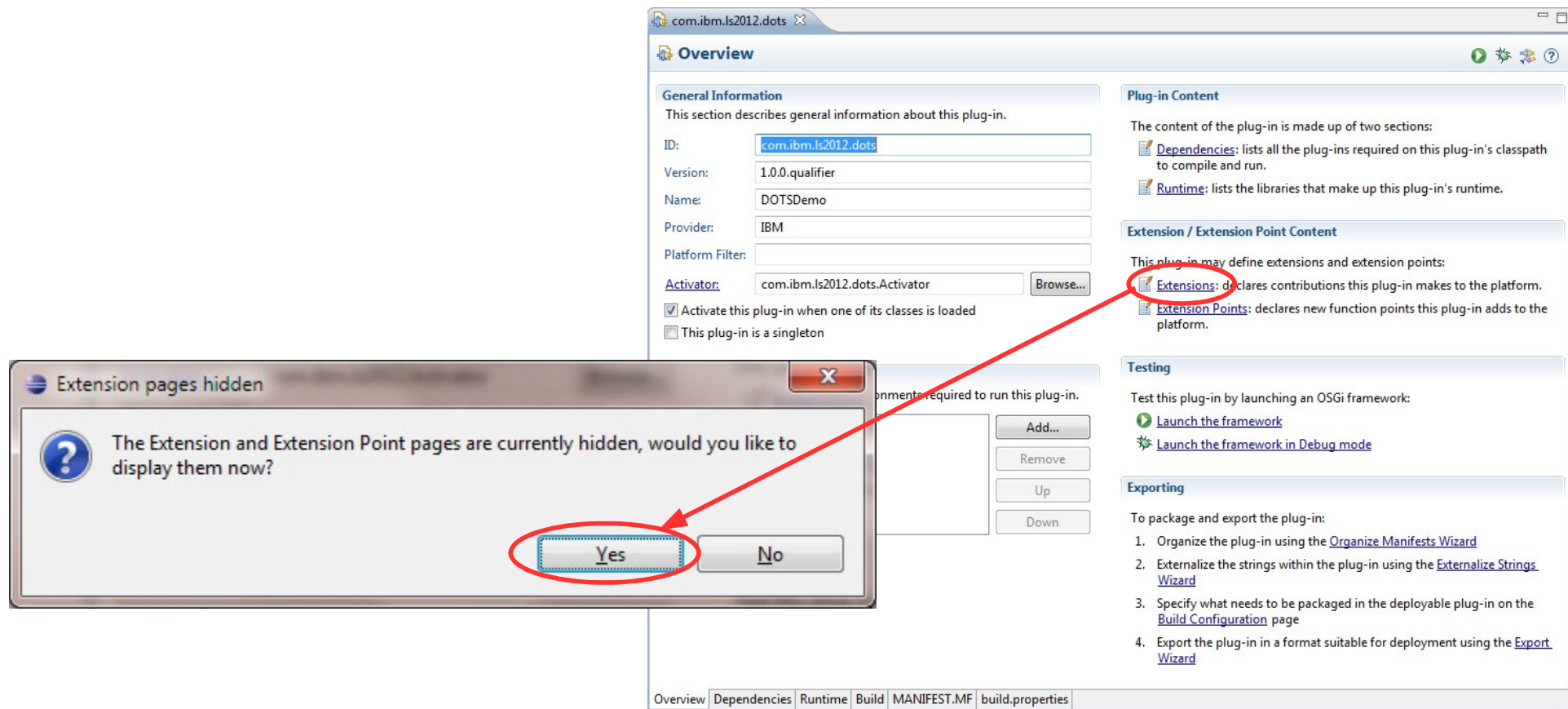
Creating DOTS tasklets

- Project Name: com.ibm.ls2012.dots
 - Select “an OSGi framework” radio button under the Target Platform section
 - Click on the Next button
- On the Content screen change the name to DOTSDemo
 - Click on the Finish button



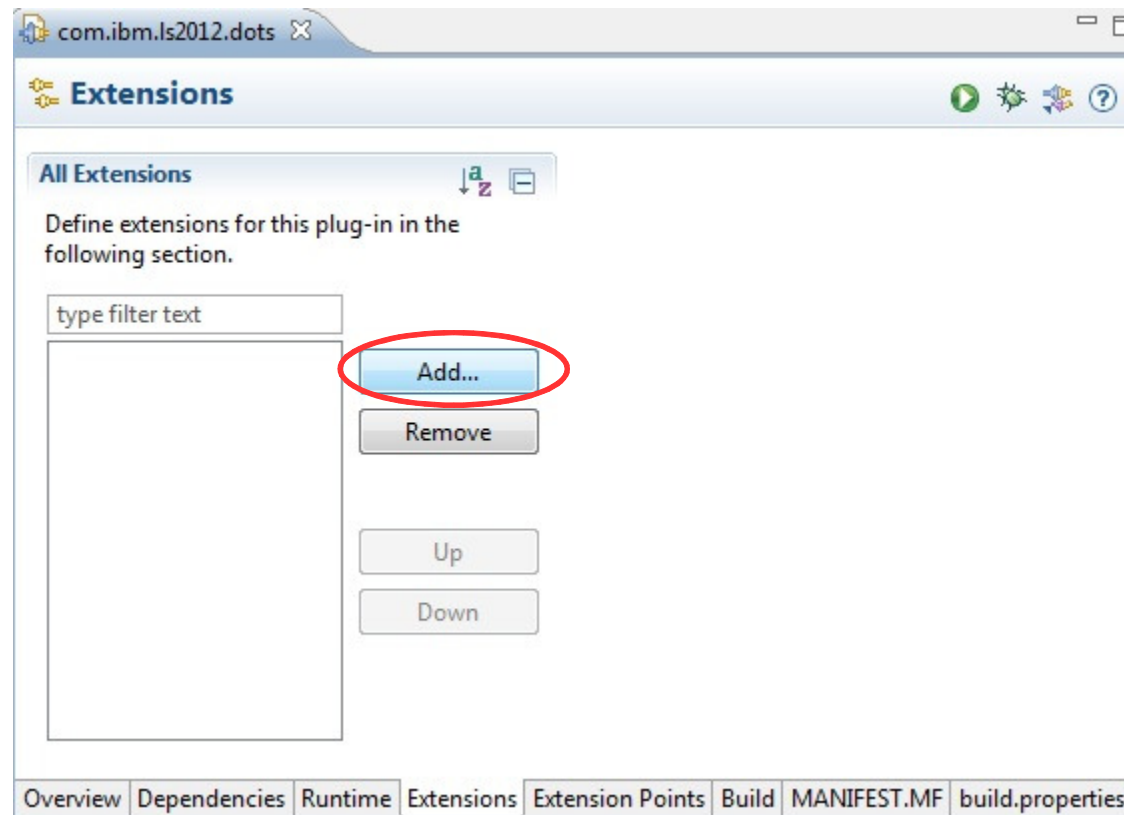
Creating DOTS tasklets

- Go to the Overview tab, under Extension / Extension Point Content and click on Extensions
- Click Yes on the Extensions pages hidden screen



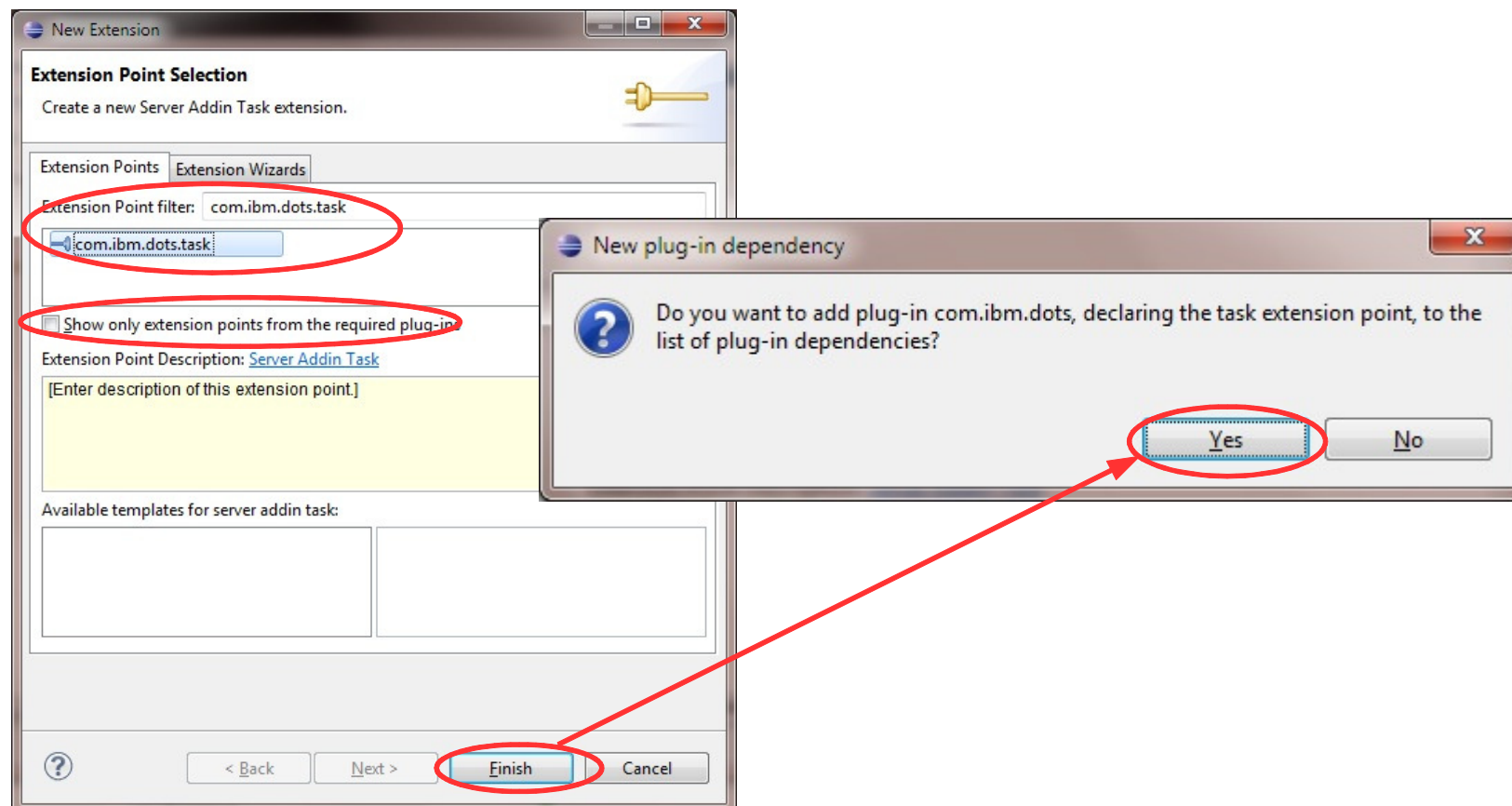
Creating DOTS tasklets

- Go to the Extensions tab and click the Add button...



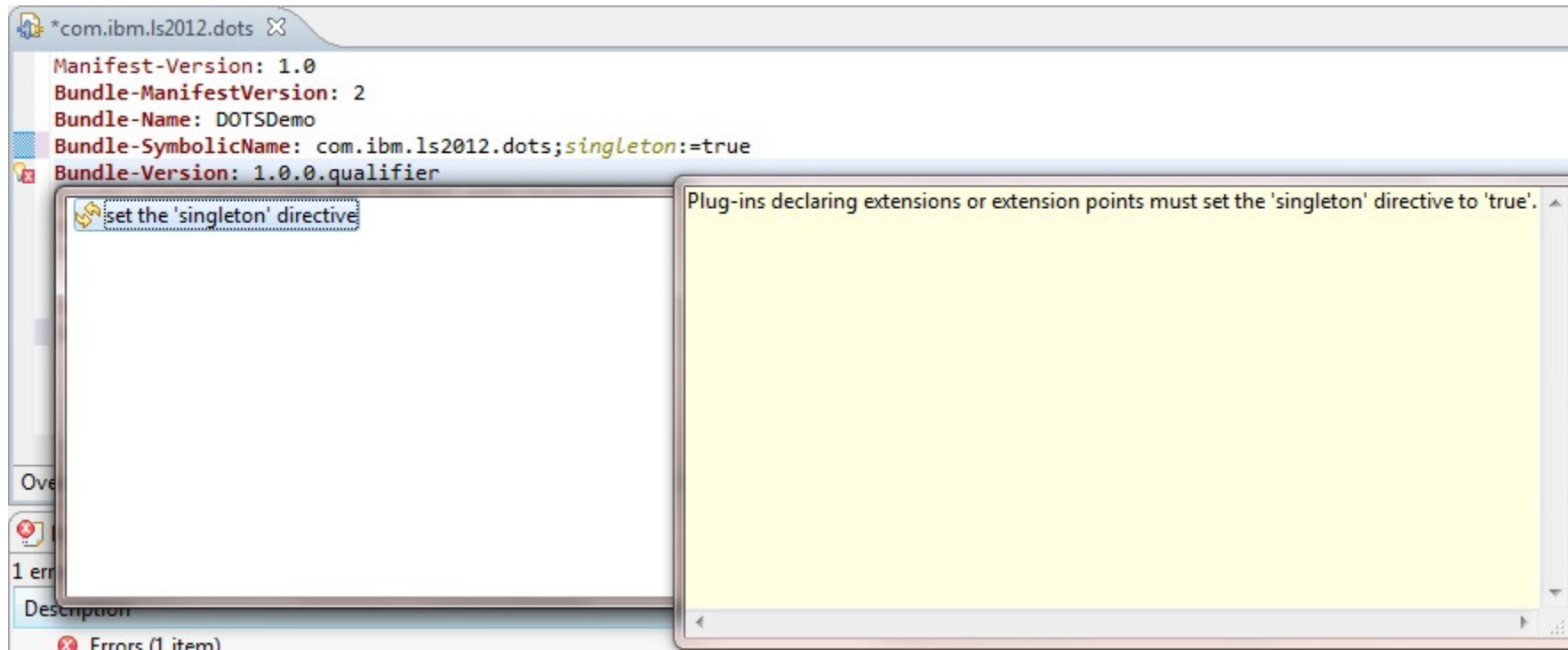
Creating DOTS tasklets

- Deselect the “Show only extension points from the required plug-ins”
 - Select the com.ibm.dots.task Extension Point
 - Click on the Finish button
- On the New plug-in dependency screen, click the Yes button



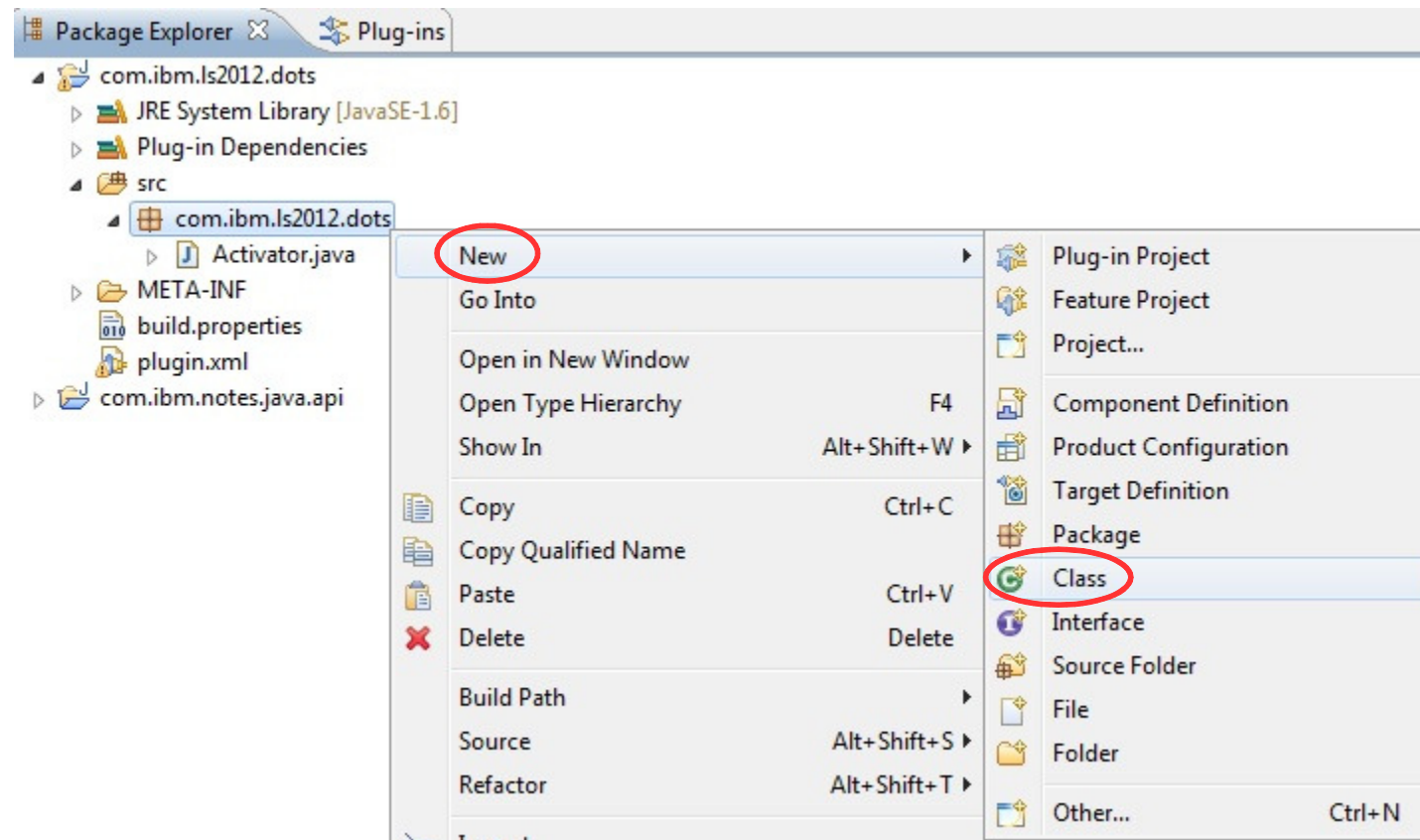
Creating DOTS tasklets

- Go to the MANIFEST.MF tab and click on the error and fix the singleton



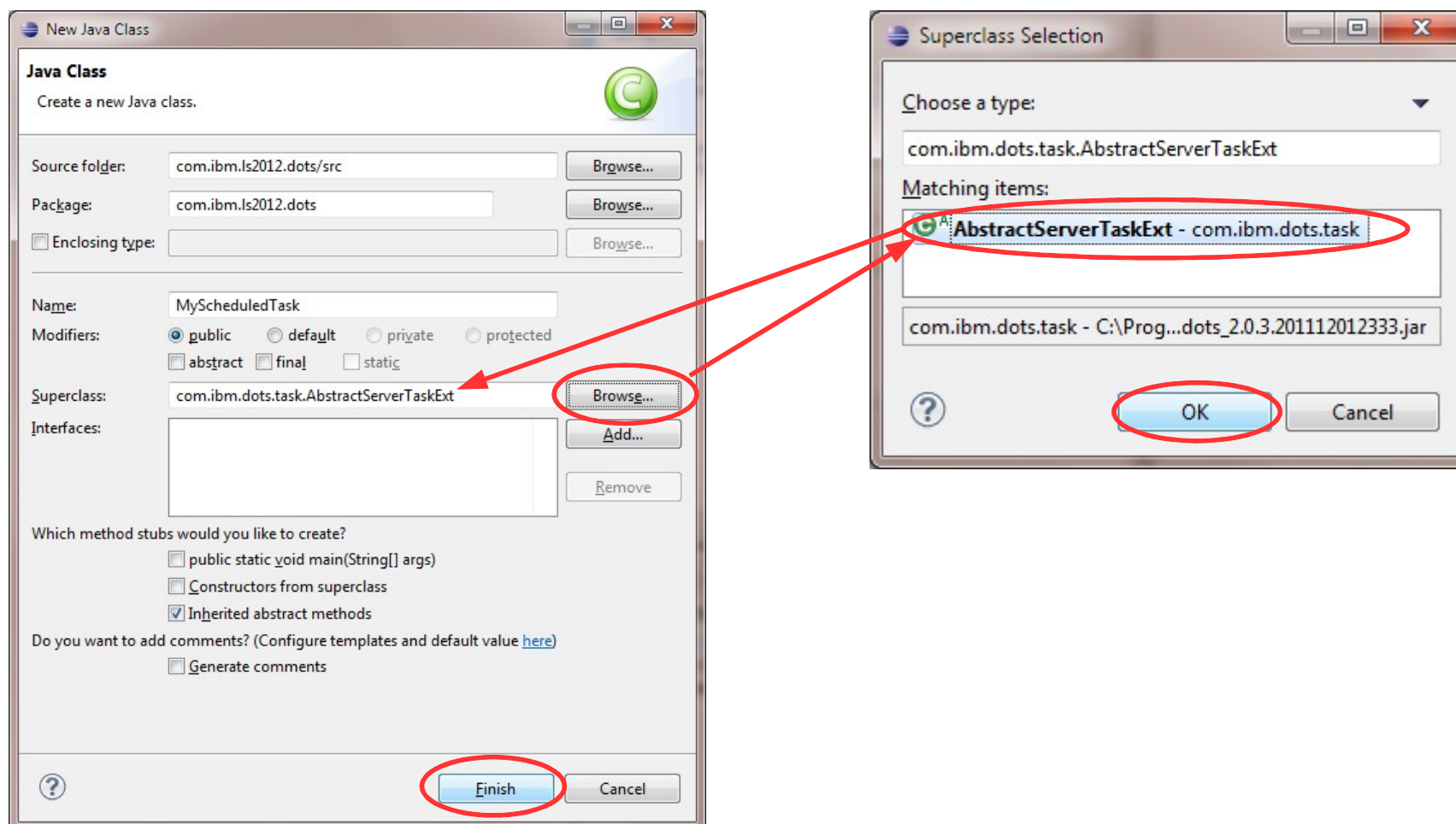
Creating DOTS tasklets

- Right click on the com.ibm.ls2012.dots package and click New->Class



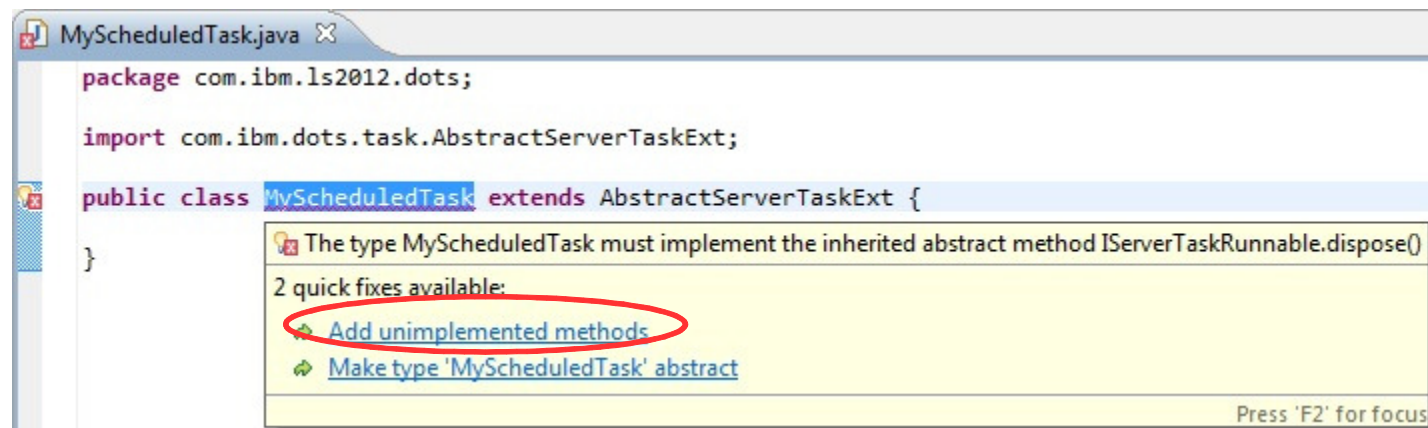
Creating DOTS tasklets

- Create a new Java class named MyScheduledTask extending com.ibm.dots.task.AbstractServerTaskExt



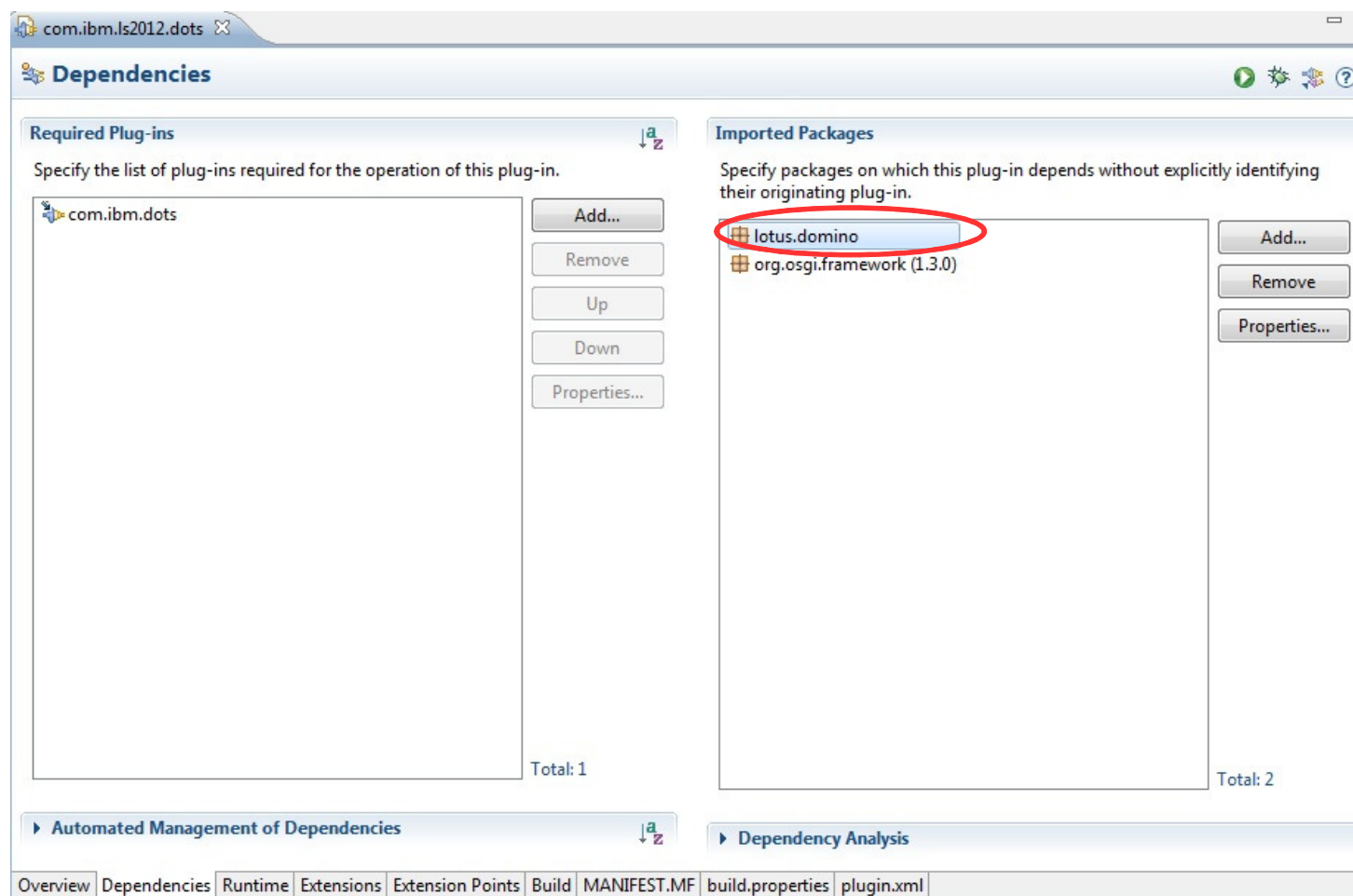
Creating DOTS tasklets

- In the MyScheduledTask class add unimplemented methods



Creating DOTS tasklets

- Resolving some of the warnings and errors displayed
 - On the Dependencies tab of MANIFEST.MF add lotus.domino to the list of Imported Packages



Creating DOTS tasklets

- Scheduling of the MyScheduledTask tasklet via plugin.xml

The screenshot displays two files in the Eclipse IDE:

MyScheduledTask.java

```
package com.ibm.ls2012.dots;

import lotus.domino.NotesException;

import org.eclipse.core.runtime.IProgressMonitor;

import com.ibm.dots.task.AbstractServerTaskExt;
import com.ibm.dots.task.RunWhen;

public class MyScheduledTask extends AbstractServerTaskExt {

    @Override
    public void dispose() throws NotesException {
        // TODO Auto-generated method stub
    }

    @Override
    protected void doRun(RunWhen arg0, IProgressMonitor arg1)
        throws NotesException {
        logMessage("Lotusphere 2012 - DOTS running MyScheduledTask via plugin.xml");
    }
}
```

plugin.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<?eclipse version="3.4"?>
<plugin>
    <extension
        point="com.ibm.dots.task">
        <task
            class="com.ibm.ls2012.dots.MyScheduledTask"
            id="com.ibm.ls2012.dots.myscheduledtask"
            description="Scheduled Demo Task (30 Second)">
            <run every="30" unit="second"/>
        </task>
    </extension>
</plugin>
```

A red arrow points from the `MyScheduledTask` class in the Java file to the `class="com.ibm.ls2012.dots.MyScheduledTask"` attribute in the XML file, illustrating the mapping between the code and the plugin configuration.

Creating DOTS tasklets

- Scheduling of the MyScheduledTaskAnnotated tasklet via annotations
 - Custom method using annotations
 - Tasklet still needs to be registered in plugin.xml

```
MyScheduledTaskAnnotated.java
package com.ibm.ls2012.dots;

import lotus.domino.NotesException;

import org.eclipse.core.runtime.IProgressMonitor;

import com.ibm.dots.annotation.RunEvery;
import com.ibm.dots.task.AbstractServerTaskExt;
import com.ibm.dots.task.RunWhen;
import com.ibm.dots.task.RunWhen.RunUnit;

public class MyScheduledTaskAnnotated extends AbstractServerTaskExt {

    @Override
    public void dispose() throws NotesException {
        // TODO Auto-generated method stub
    }

    @Override
    protected void doRun(RunWhen arg0, IProgressMonitor arg1)
        throws NotesException {

    }

    @RunEvery(every=20, unit=RunUnit.second)
    public void myDoRunMethod(IProgressMonitor monitor)
        throws NotesException {
        logMessage("Lotusphere 2012 - DOTS running MyScheduledTaskAnnotated via annotations");
    }
}

plugin.xml
<?xml version="1.0" encoding="UTF-8"?>
<?eclipse version="3.4"?>
<plugin>
    <extension
        point="com.ibm.dots.task">
        <task
            class="com.ibm.ls2012.dots.MyScheduledTask"
            id="com.ibm.ls2012.dots.myscheduledtask"
            description="Scheduled Demo Task (30 Second)">
            <run every="30" unit="second"/>
        </task>
        <task
            class="com.ibm.ls2012.dots.MyScheduledTaskAnnotated"
            id="com.ibm.ls2012.dots.myscheduledtaskannotated">
        </task>
        </extension>
    </plugin>
```


Creating DOTS tasklets

- MyManualTask tasklet with the final plugin.xml

```
MyManualTask.java X
package com.ibm.ls2012.dots;

import lotus.domino.NotesException;

import org.eclipse.core.runtime.IProgressMonitor;

import com.ibm.dots.task.AbstractServerTaskExt;
import com.ibm.dots.task.RunWhen;

public class MyManualTask extends AbstractServerTaskExt {

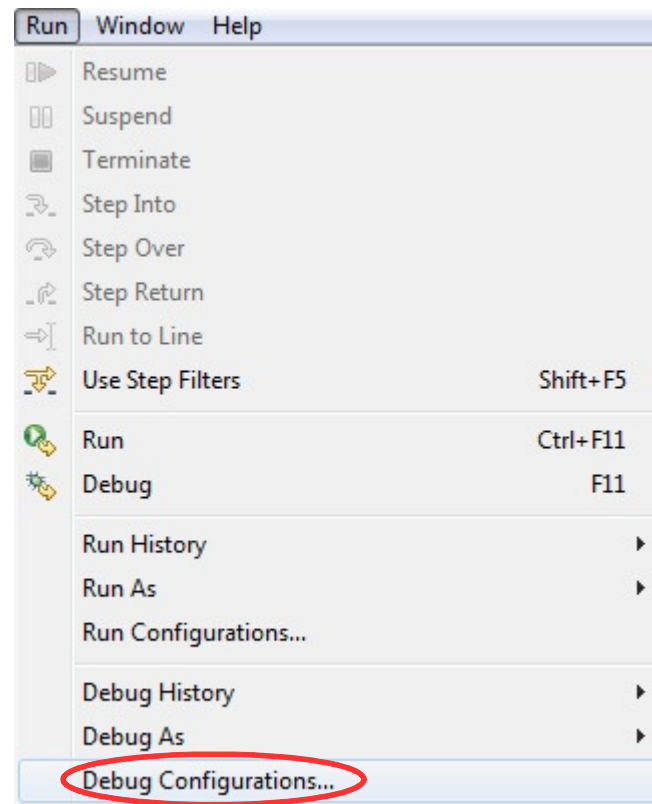
    @Override
    public void dispose() throws NotesException {
        // TODO Auto-generated method stub
    }

    @Override
    protected void doRun(RunWhen arg0, IProgressMonitor arg1)
        throws NotesException {
        logMessage("Lotusphere 2012 - DOTS running MyManualTask");
    }
}
```

```
plugin.xml X
<?xml version="1.0" encoding="UTF-8"?>
<?eclipse version="3.4"?>
<plugin>
  <extension
    point="com.ibm.dots.task">
    <task
      class="com.ibm.ls2012.dots.MyScheduledTask"
      id="com.ibm.ls2012.dots.myscheduledtask"
      description="Scheduled Demo Task (30 Second)">
      <run every="30" unit="second"/>
    </task>
    <task
      class="com.ibm.ls2012.dots.MyScheduledTaskAnnotated"
      id="com.ibm.ls2012.dots.myscheduledtaskannotated">
    </task>
    <task
      class="com.ibm.ls2012.dots.MyManualTask"
      id="com.ibm.ls2012.dots.mymanualtask"
      description="Scheduled Demo Task (Manual)"
      runOnStart="true">
    </task>
  </extension>
</plugin>
```

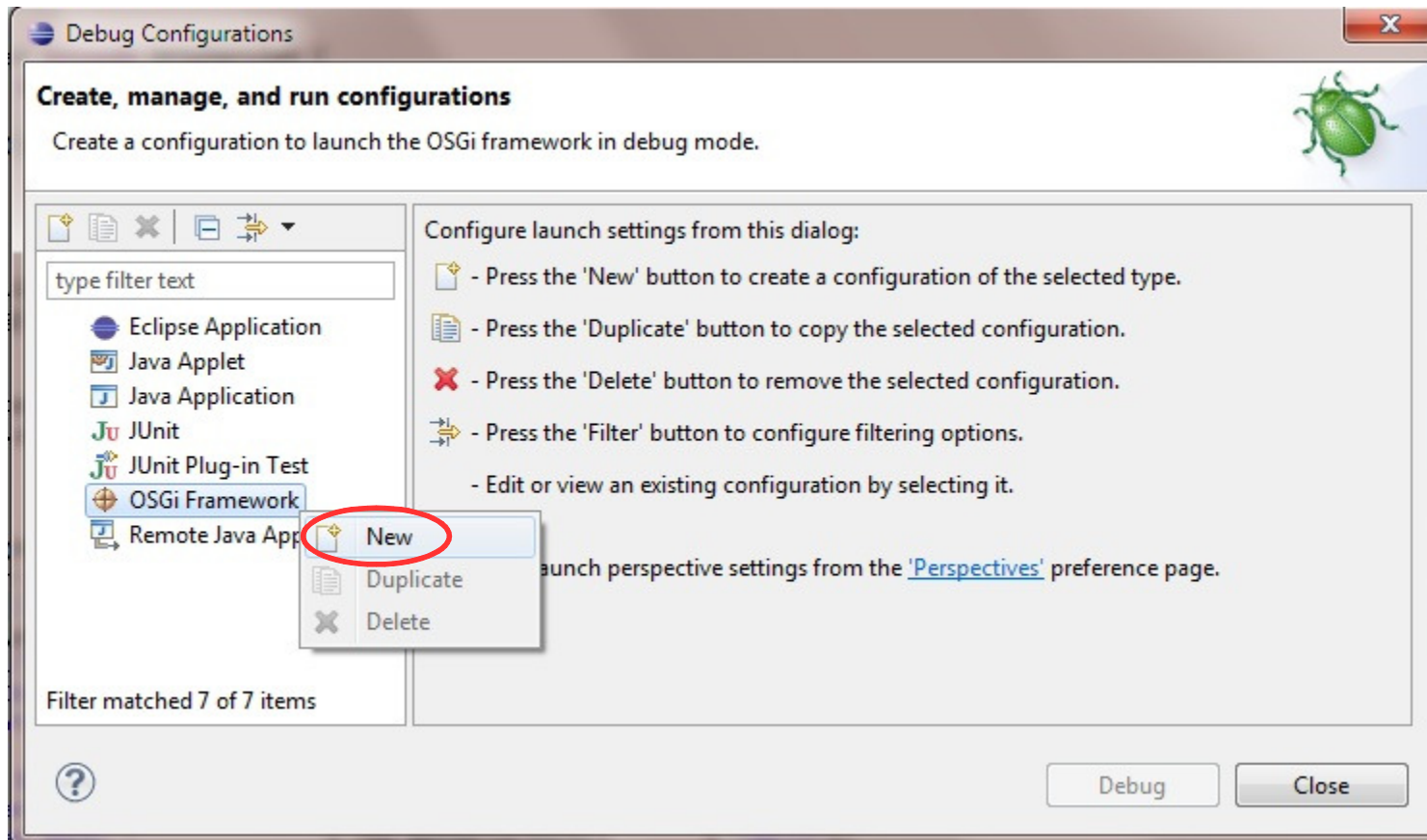
Deploying & debugging a DOTS tasklet

- Click on Run->Debug Configurations...

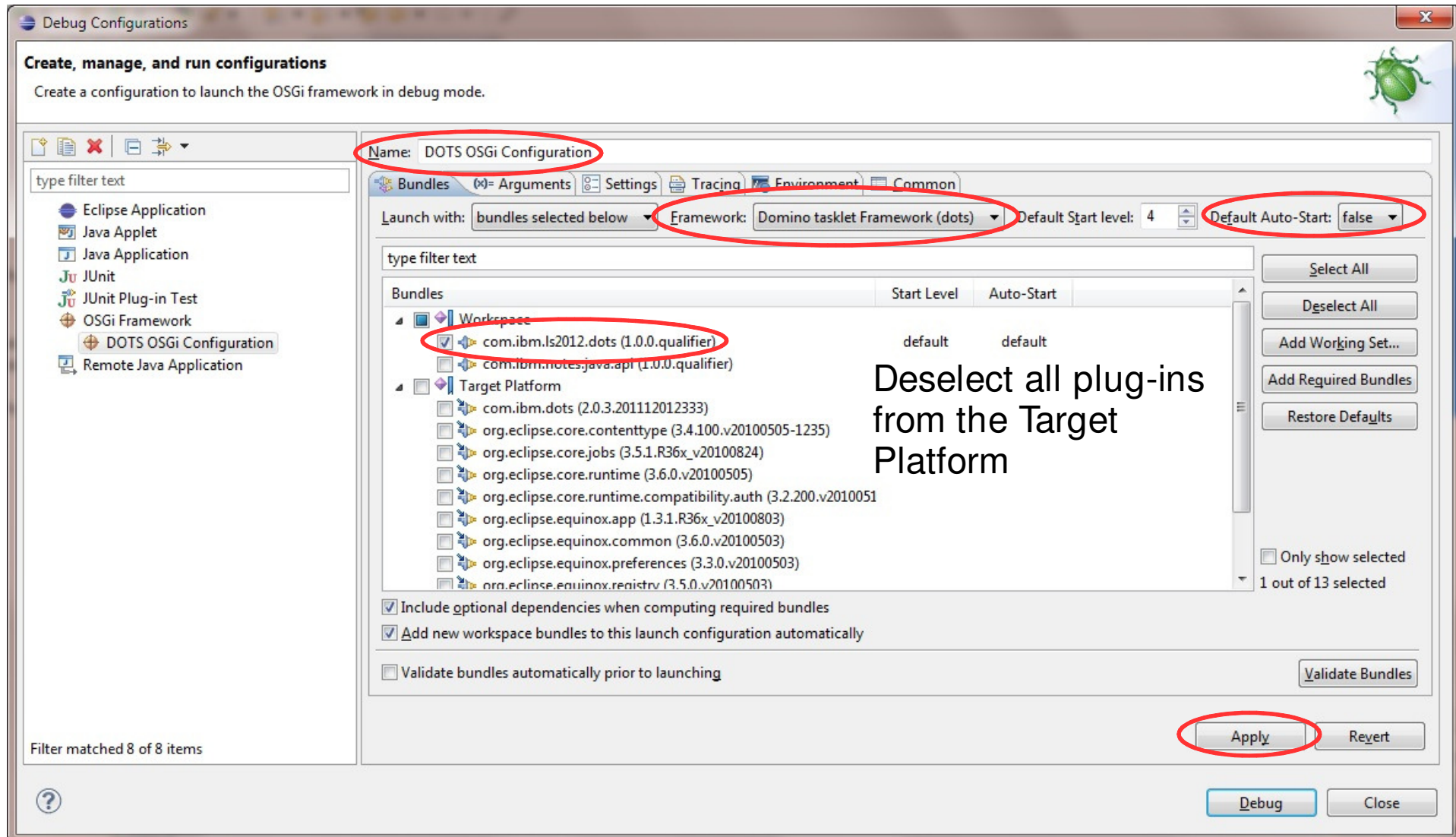


Deploying & debugging a DOTS tasklet

- Create a new OSGi Framework

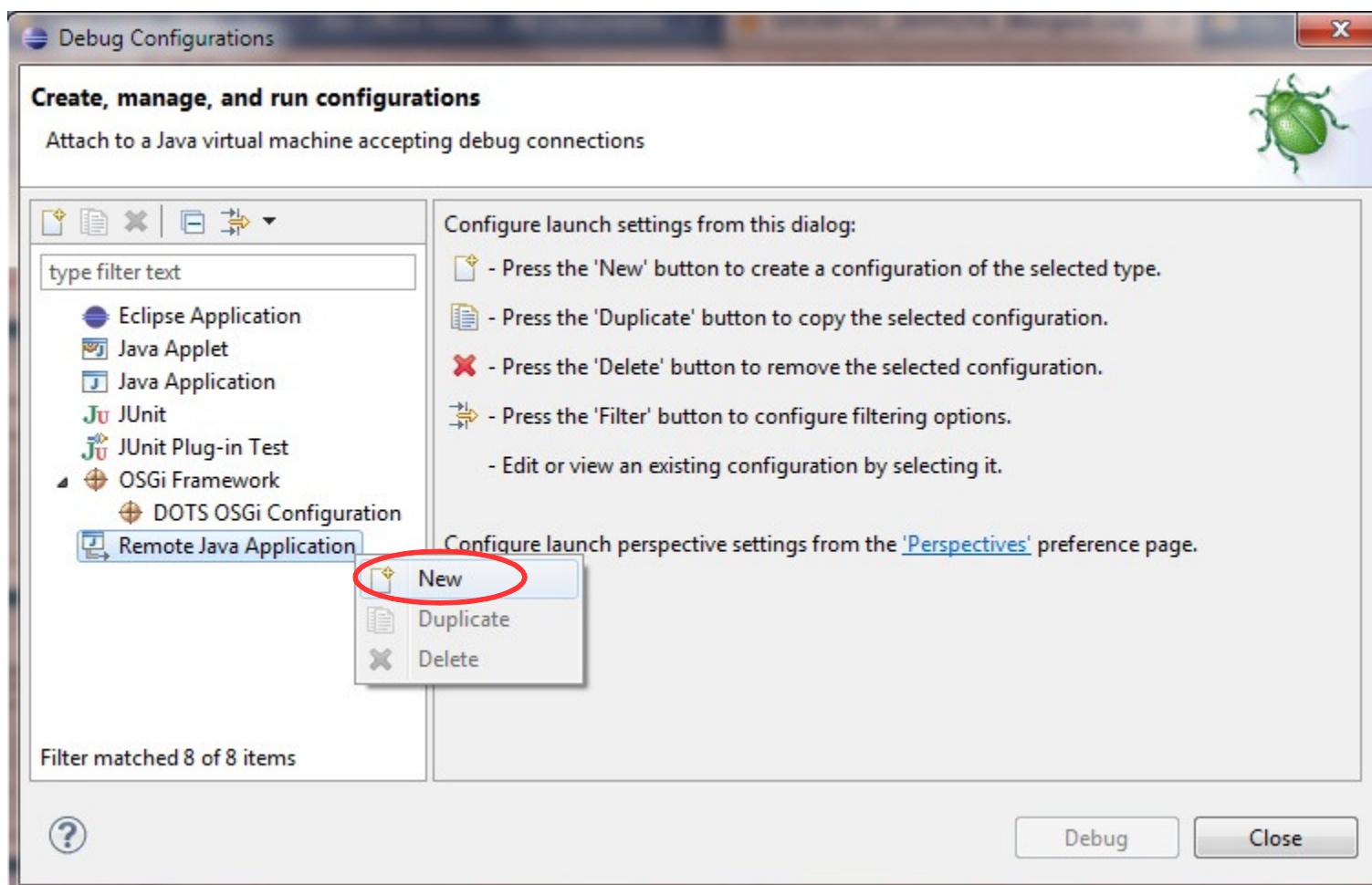


Deploying & debugging a DOTS tasklet

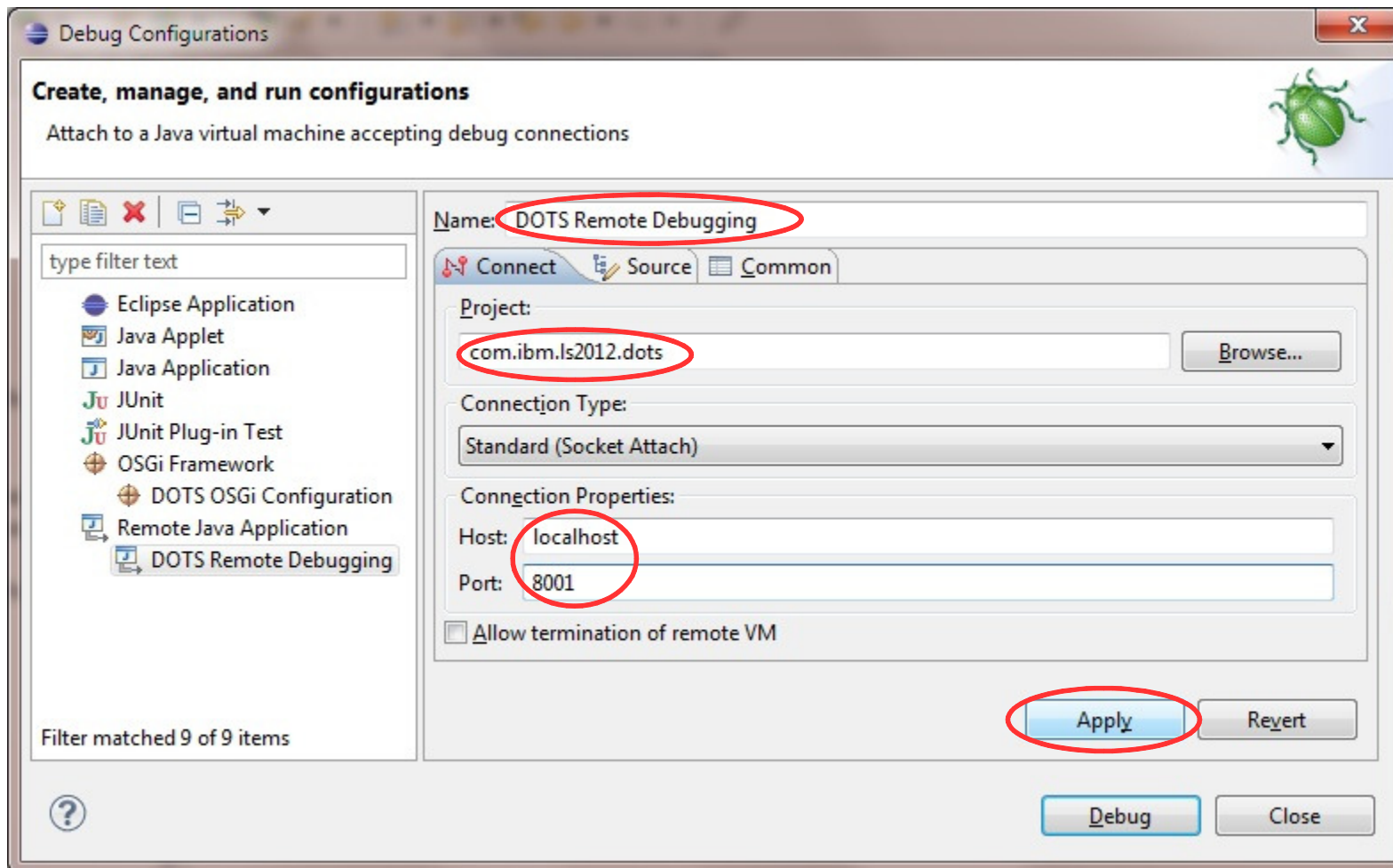


Deploying & debugging a DOTS tasklet

- Create a new Remote Java Application

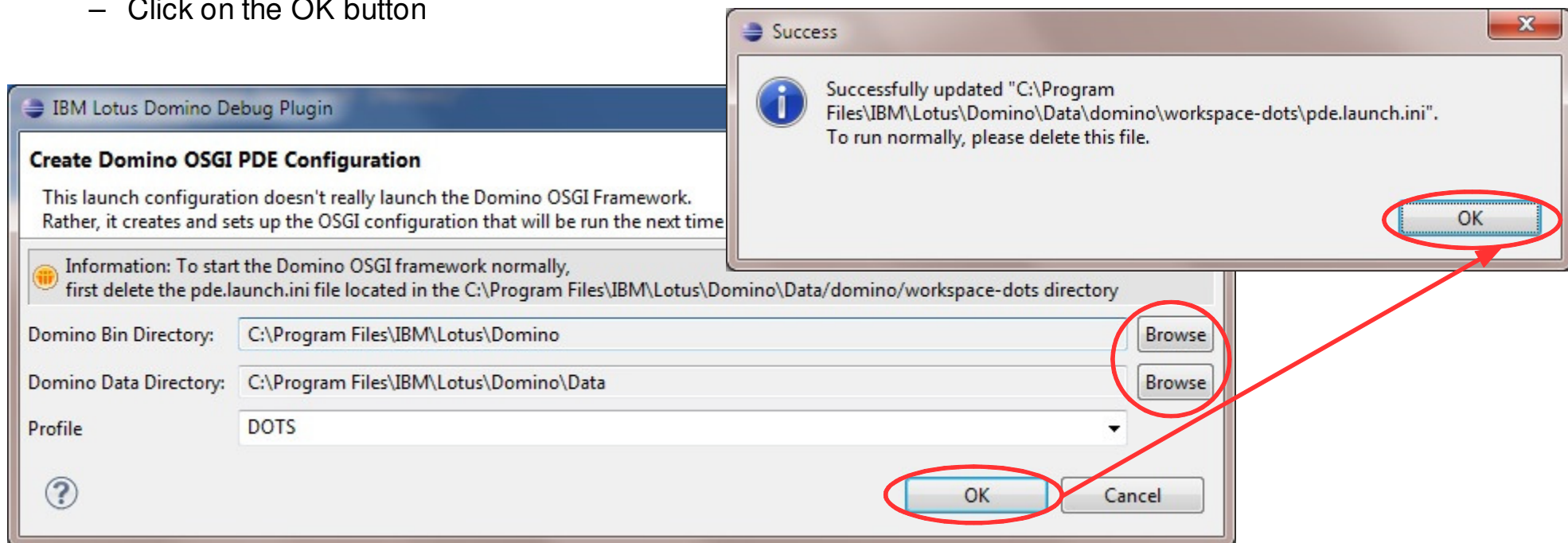


Deploying & debugging a DOTS tasklet



Deploying & debugging a DOTS tasklet

- Click on the DOTS OSGi Configuration and then click the Debug button
 - This launches the PDE Tool
 - Select the appropriate Domino Binary & Data directories
 - Keep the default Profile of DOTS
 - Click on the OK Button
- If everything worked OK then you'll see a Success dialog
 - Click on the OK button



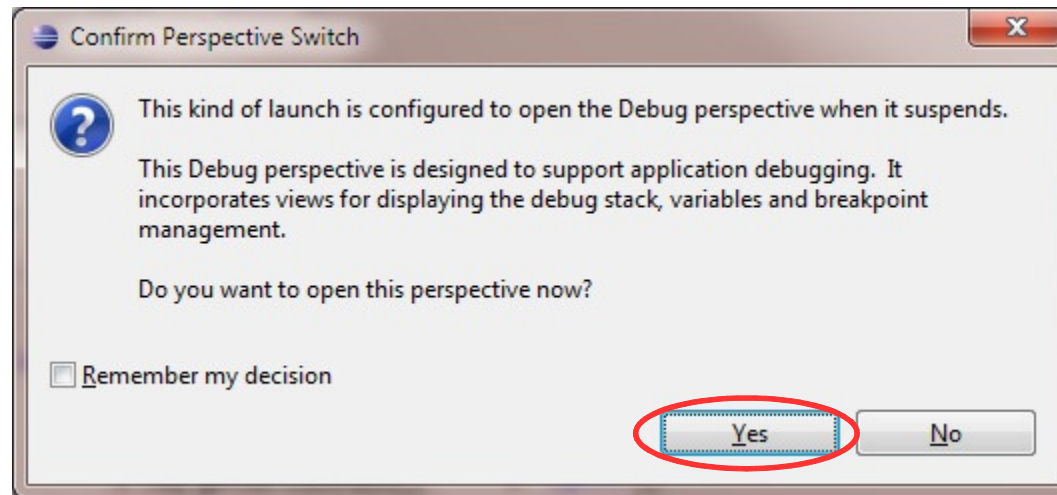
Deploying & debugging a DOTS tasklet

- Finally, time to run the tasklets and break into the debugger
- Enable debugging via notes.ini
 - DOTS_DEBUGADDRESS=8001
 - DOTS_DEBUGSUSPEND=y
- load dots on the server console
- Set a break point in your code
- Go back to Run->Debug Configurations...
 - Select DOTS Remote Debugging under Remote Java Applications
 - Click on the Debug button

```
> load dots  
> Listening for transport dt_socket at address: 8001
```

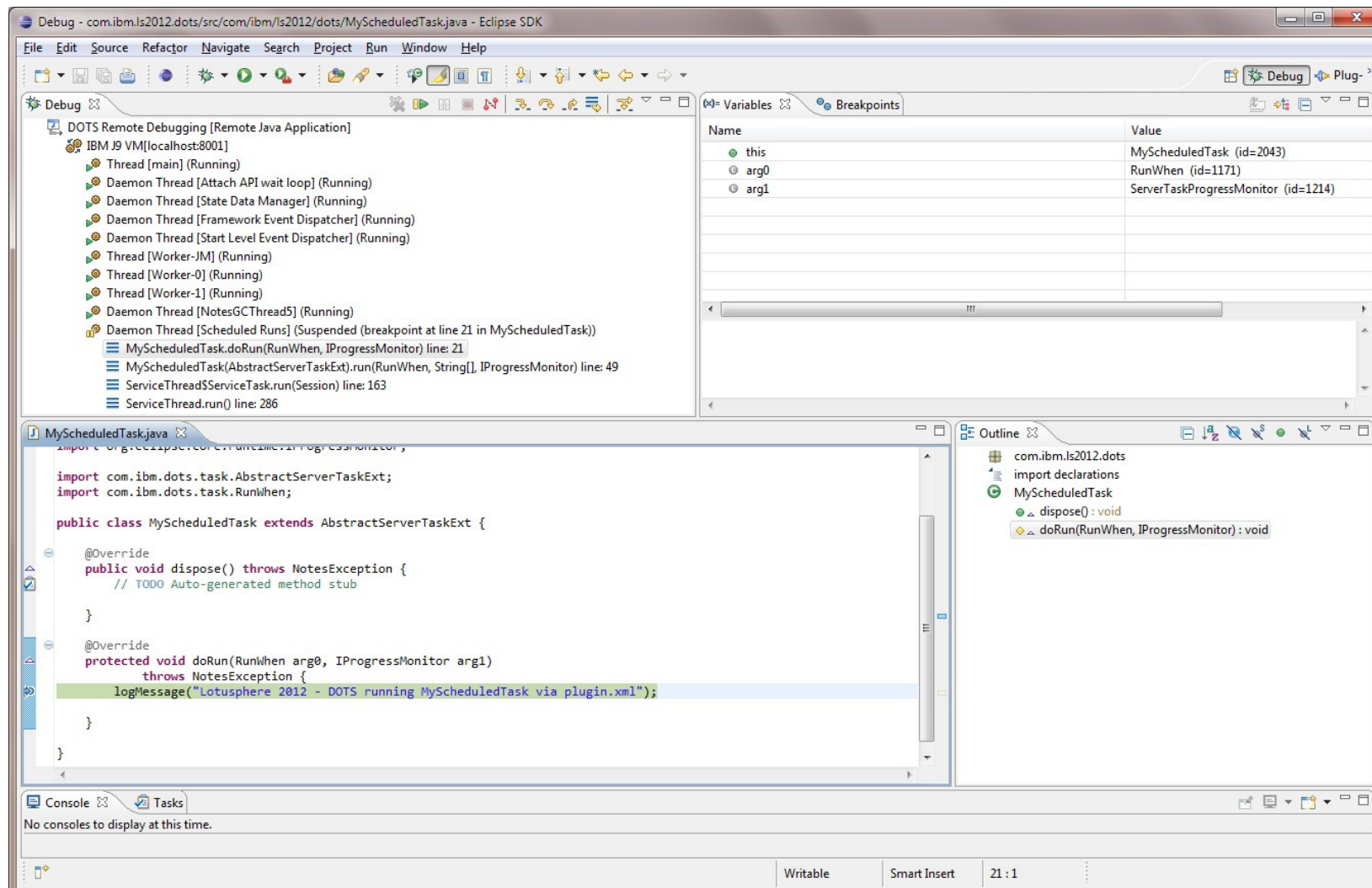

Deploying & debugging a DOTS tasklet

- Confirm the switch to the Debug Perspective by clicking the Yes button



Deploying & debugging a DOTS tasklet

- Debug as required



Deploying & debugging a DOTS tasklet

- Output examples

```
> load dots
> WARNING: Using pde configuration DOTS OSGi Configuration located in C:\Program Files\IBM\Lotus\Domino\Data\domino\work
space-dots\pde.launch.ini
[1D18:0002-1B10] 12/15/2011 05:01:13 PM Domino OSGi Tasklet Service started < profile DOTS >
[1D18:000A-19AC] 12/15/2011 05:01:15 PM [DOTS] <com.ibm.ls2012.dots.mymanualtask> Lotusphere 2012 - DOTS running MyManu
alTask
[1D18:000C-04BC] 12/15/2011 05:01:29 PM [DOTS] <com.ibm.ls2012.dots.myscheduledtask> Lotusphere 2012 - DOTS running MyS
cheduledTask via plugin.xml
[1D18:000C-04BC] 12/15/2011 05:01:29 PM [DOTS] <com.ibm.ls2012.dots.myscheduledtaskannotated> Lotusphere 2012 - DOTS ru
nning MyScheduledTaskAnnotated via annotations
[1D18:000C-04BC] 12/15/2011 05:01:49 PM [DOTS] <com.ibm.ls2012.dots.myscheduledtaskannotated> Lotusphere 2012 - DOTS ru
nning MyScheduledTaskAnnotated via annotations
[1D18:000C-04BC] 12/15/2011 05:01:59 PM [DOTS] <com.ibm.ls2012.dots.myscheduledtask> Lotusphere 2012 - DOTS running MyS
cheduledTask via plugin.xml
tell dots quit
[1D18:0002-1B10] 12/15/2011 05:02:05 PM Domino OSGi Tasklet Service terminated < profile DOTS >
```

```
> tell dots tasklist
[1B54:0002-1C34] 12/15/2011 05:08:36 PM [DOTS]
[1B54:0002-1C34] 12/15/2011 05:08:36 PM [DOTS] id:      com.ibm.ls2012.dots.myscheduledtask
[1B54:0002-1C34] 12/15/2011 05:08:36 PM [DOTS] description:  Scheduled Demo Task (30 Second)
[1B54:0002-1C34] 12/15/2011 05:08:36 PM [DOTS] Run on Start:  false
[1B54:0002-1C34] 12/15/2011 05:08:36 PM [DOTS] Runs every 30 - second
[1B54:0002-1C34] 12/15/2011 05:08:36 PM [DOTS]
[1B54:0002-1C34] 12/15/2011 05:08:36 PM [DOTS] id:      com.ibm.ls2012.dots.myscheduledtaskannotated
[1B54:0002-1C34] 12/15/2011 05:08:36 PM [DOTS] description:  null
[1B54:0002-1C34] 12/15/2011 05:08:36 PM [DOTS] Run on Start:  false
[1B54:0002-1C34] 12/15/2011 05:08:36 PM [DOTS] Runs every 20 - second
[1B54:0002-1C34] 12/15/2011 05:08:36 PM [DOTS]
[1B54:0002-1C34] 12/15/2011 05:08:36 PM [DOTS] id:      com.ibm.ls2012.dots.mymanualtask
[1B54:0002-1C34] 12/15/2011 05:08:36 PM [DOTS] description:  Scheduled Demo Task (Manual)
[1B54:0002-1C34] 12/15/2011 05:08:36 PM [DOTS] Run on Start:  true
[1B54:0002-1C34] 12/15/2011 05:08:36 PM [DOTS] Not scheduled
```

```
> tell dots ss ls2012
[1B54:0002-1C34] 12/15/2011 05:08:58 PM [DOTS]
[1B54:0002-1C34] 12/15/2011 05:08:58 PM [DOTS] Framework is launched.
[1B54:0002-1C34] 12/15/2011 05:08:58 PM [DOTS]
[1B54:0002-1C34] 12/15/2011 05:08:58 PM [DOTS] id
[1B54:0002-1C34] 12/15/2011 05:08:58 PM [DOTS] State      Bundle
[1B54:0002-1C34] 12/15/2011 05:08:58 PM [DOTS]
[1B54:0002-1C34] 12/15/2011 05:08:58 PM [DOTS] 10      ACTIVE      com.ibm.ls2012.dots_1.0.0.qualifier
```

Questions or Comments?

Legal disclaimer

© IBM Corporation 2012. All Rights Reserved.

The information contained in this publication is provided for informational purposes only. While efforts were made to verify the completeness and accuracy of the information contained in this publication, it is provided AS IS without warranty of any kind, express or implied. In addition, this information is based on IBM's current product plans and strategy, which are subject to change by IBM without notice. IBM shall not be responsible for any damages arising out of the use of, or otherwise related to, this publication or any other materials. Nothing contained in this publication is intended to, nor shall have the effect of, creating any warranties or representations from IBM or its suppliers or licensors, or altering the terms and conditions of the applicable license agreement governing the use of IBM software.

References in this presentation to IBM products, programs, or services do not imply that they will be available in all countries in which IBM operates. Product release dates and/or capabilities referenced in this presentation may change at any time at IBM's sole discretion based on market opportunities or other factors, and are not intended to be a commitment to future product or feature availability in any way. Nothing contained in these materials is intended to, nor shall have the effect of, stating or implying that any activities undertaken by you will result in any specific sales, revenue growth or other results.

IBM, the IBM logo, Lotus, Lotus Notes, Notes, Domino, Quickr, Sametime, WebSphere, UC2, PartnerWorld and Lotusphere are trademarks of International Business Machines Corporation in the United States, other countries, or both. Unyte is a trademark of WebDialogs, Inc., in the United States, other countries, or both.

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.