



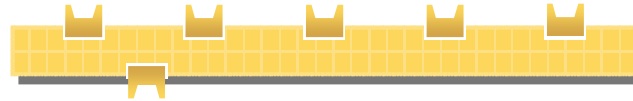
DEV-4: Extending Progress Dynamics®

Tom Greenwood
Degama Systems Inc.
President, GTA-PUG
tomg@degama.com

Agenda

- Dynamics and the OpenEdge® Reference Architecture
- Managers and their uses
- Some Useful Customization Examples
 1. Report Options
 2. Dynamic Lookup By Keyword
 3. Dynamic Browse Enhancements
 4. DCU Deployment
- Summary of Methods of Customizing
- Conclusions

OpenEdge RA and Dynamics



Rendering Programs

Enterprise Services

Procedure Libraries ('PLIPs')

SDO's

Data Access

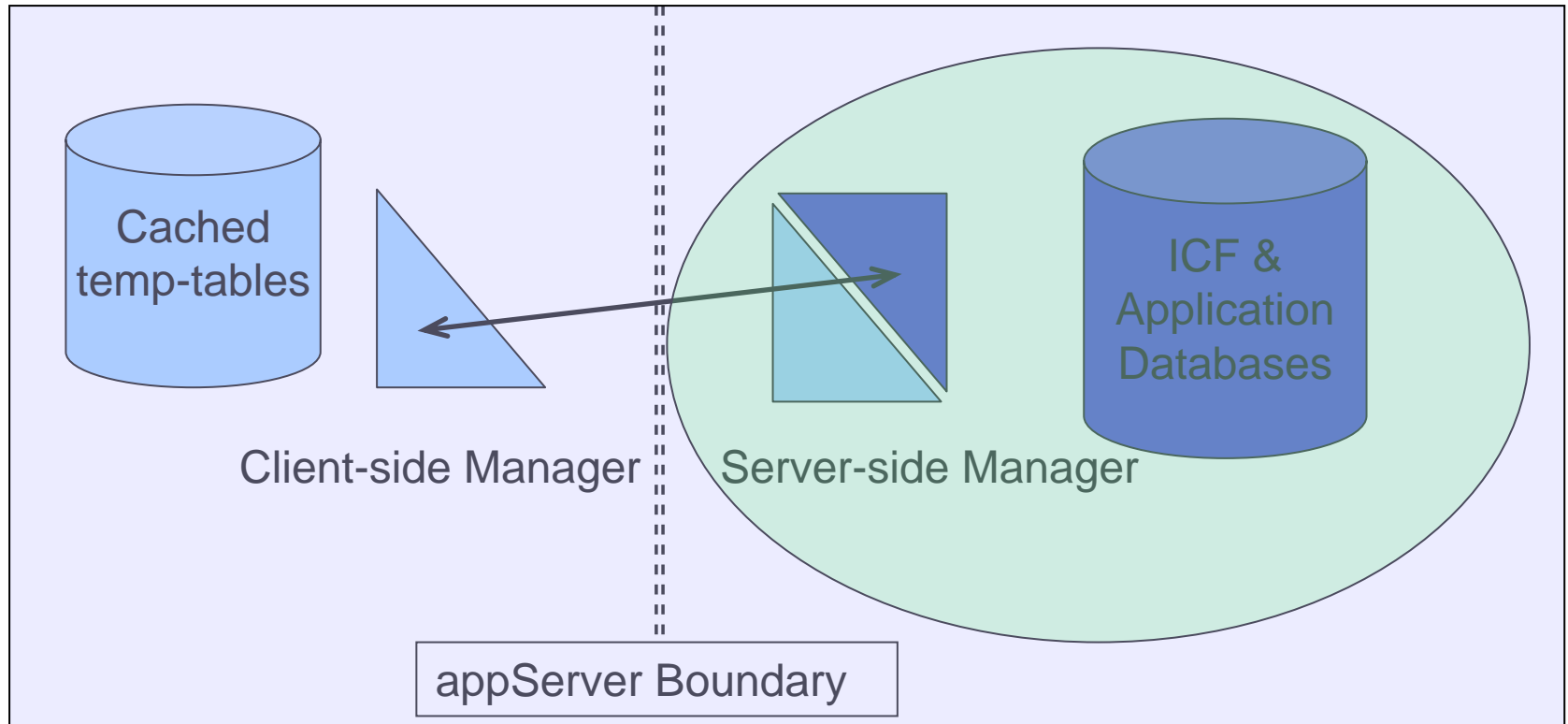
Data Sources

**Common
Infrastructure**

Managers

- * Session
- * General
- * Security
- * Profile
- * Localization
- * Customization

Anatomy of a manager



```
/* dpstsclnp.p */  
&global-define client-side  
yes  
{tr/bl/dpstsmngrp.i}
```

```
/* dpstssrvrp.p */  
&global-define server-side  
yes  
{tr/bl/dpstsmngrp.i}
```

Conditional use of temp-table

```
&IF DEFINED(server-side) = 0 &THEN
    &GLOBAL-DEFINE TT          tt
    &GLOBAL-DEFINE dpStatus ttDPStatus
&Else
    &GLOBAL-DEFINE dpStatus trostat
&endif
```

```
FIND FIRST {&dpstatus}
WHERE {&dpstatus}.company = cCompany
AND {&dpstatus}.rec-status = pcStatus NO-LOCK NO-ERROR.
IF AVAILABLE {&dpstatus} THEN
    RETURN {&dpstatus}.all-Load.
ELSE
    RETURN ?.
```

Dynamics Manager

- Has 2 versions, one for client, one for server
- Server side has database access
- Client side has its data cached in temp-tables
- Client-side data may either be cached on an “as-needed” basis while applications are run, or all client-side data cached once at session start-up
- Referenced by a handle variable or you can set yours to be a session superprocedure.

Application Manager

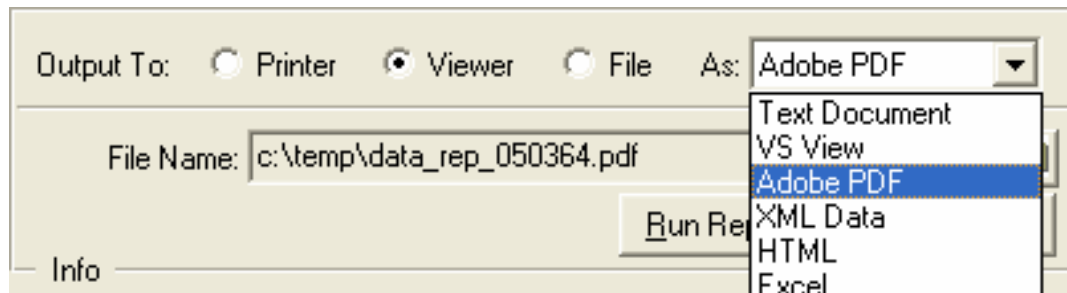
- Dynamics' Managers don't address specific application needs
- Build your own manager for:
 - Common functions and routines
 - Status codes and events
 - Static data, combo data, etc.
- Must be added to session startups
- Best way is to register in repository and use session type to export XML startup

Profile Manager

- “Remembers” each user’s preferences
- Contains user data indexed by profile type, code, and key that uniquely identifies the context of the data string
- Example for browse column positions & sizes:
 - Type = “Browser”
 - Code = “Columns”
 - Key = containerName + BrowserName
 - Data = FieldName(4)Width(3)Name(4)Width(3)
- Data only has meaning within the context provided.
- Readily customizable

Example: Extending the Framework

- Report Output From The Browse
- Dynamics Provides for a startup parameter –
 - Why not let the user choose at run time?
 - What about general reporting?
- Store user's preference via the profile manager




Report Output Enhancement

Problem:

- Enable users to select which fields they want to report on
- Enable users to identify the type of output – text, report, XML, HTML, PDF, etc.

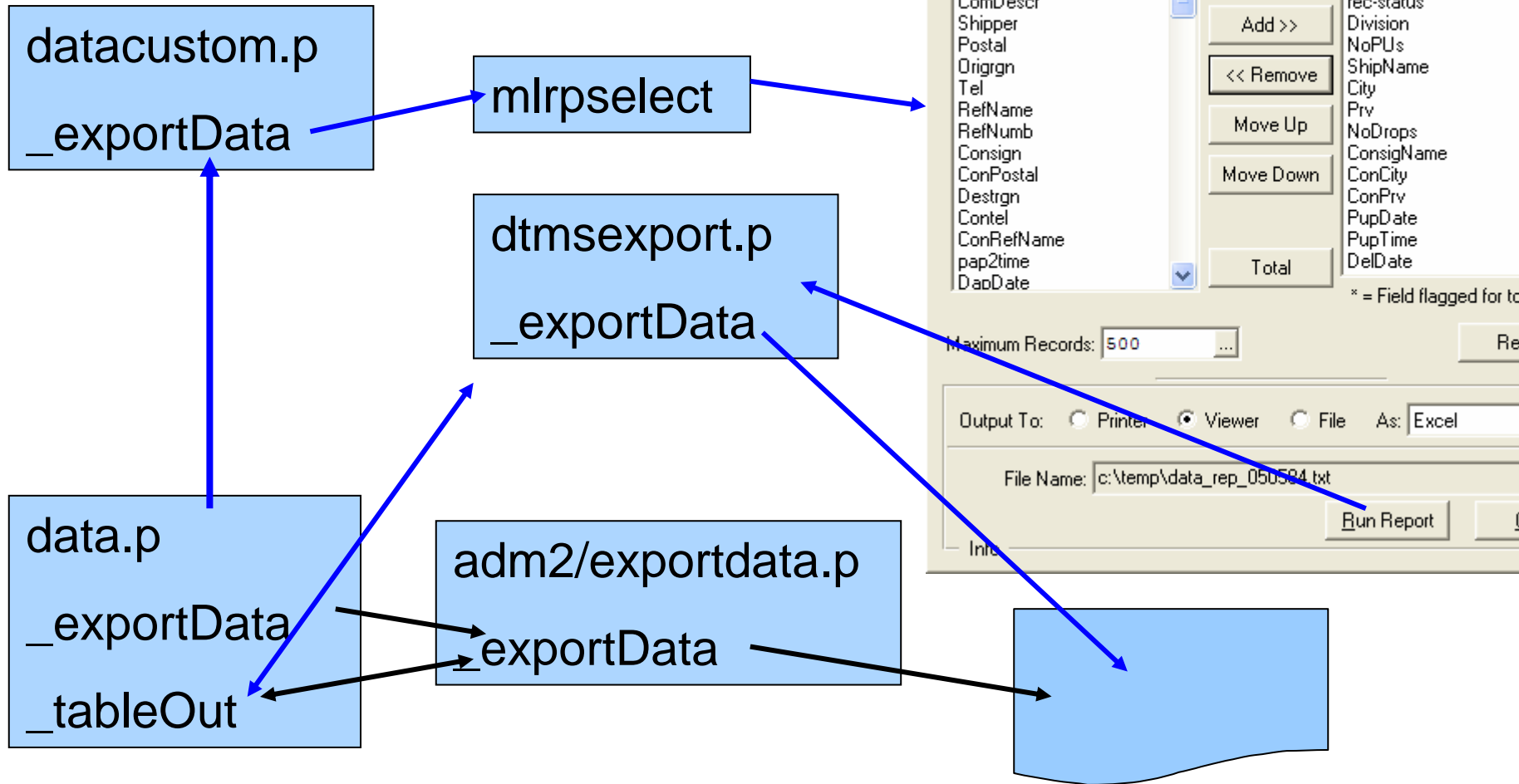
Methodology:

- Look for and trace the action on click of report button on toolbar:  (takes a little digging)
- Find the override point (exportData_data.p)
- Create (or modify) custom super (datacustom.p)

Demo



Custom data export



Future Plans

- Allow the user to name and save custom report output
- Use a data explorer tool to generate XML Datasets (from ProDataSets)
- Use XML Dataset as a data source for multiple report types: html, txt, pdf, Crystal, Excel, etc.

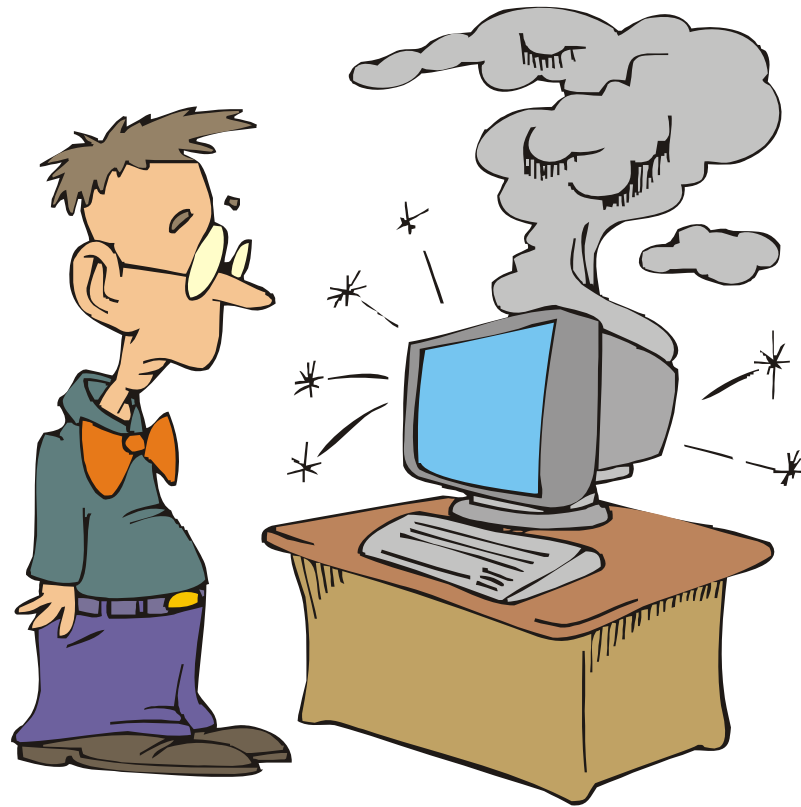
Keyword Lookup

- Standard lookup allows for “begins with” searching on a key field.
- For selecting from 1000’s of records, this is not sufficient.
- Why not have a lookup that searches a keyword index with the “contains” verb?
- Found out someone had already implemented it (Wouter Dupre)
- Caution! Not easy to port to other databases.

Keyword Lookup Implementation

- New attribute: lookupCharRelation
- Added directly to dynlookup (vs. subclassed)
- Modified rylookupfv.w, rysdflkupv.w, rysdfsuprp.p, afitlkctrl.i, lookup.p, lookupprop.i, lookupprto.i, rylookupbrv.w
- 2.1A → 2.1B upgrade was difficult due to new API; source control (RTB) critical to track customized objects in this case.

Demo



Browse Configuration

- Problem
 - “TMI” – too much information!
 - Different users, doing different things, wanted to see ONLY those fields that concerned the task they were doing (Dispatch, Billing, Brokerage, Containers, etc.)
- Method
 - Use similar interface to drop/add fields and name these “views”.

Browse Configuration Method

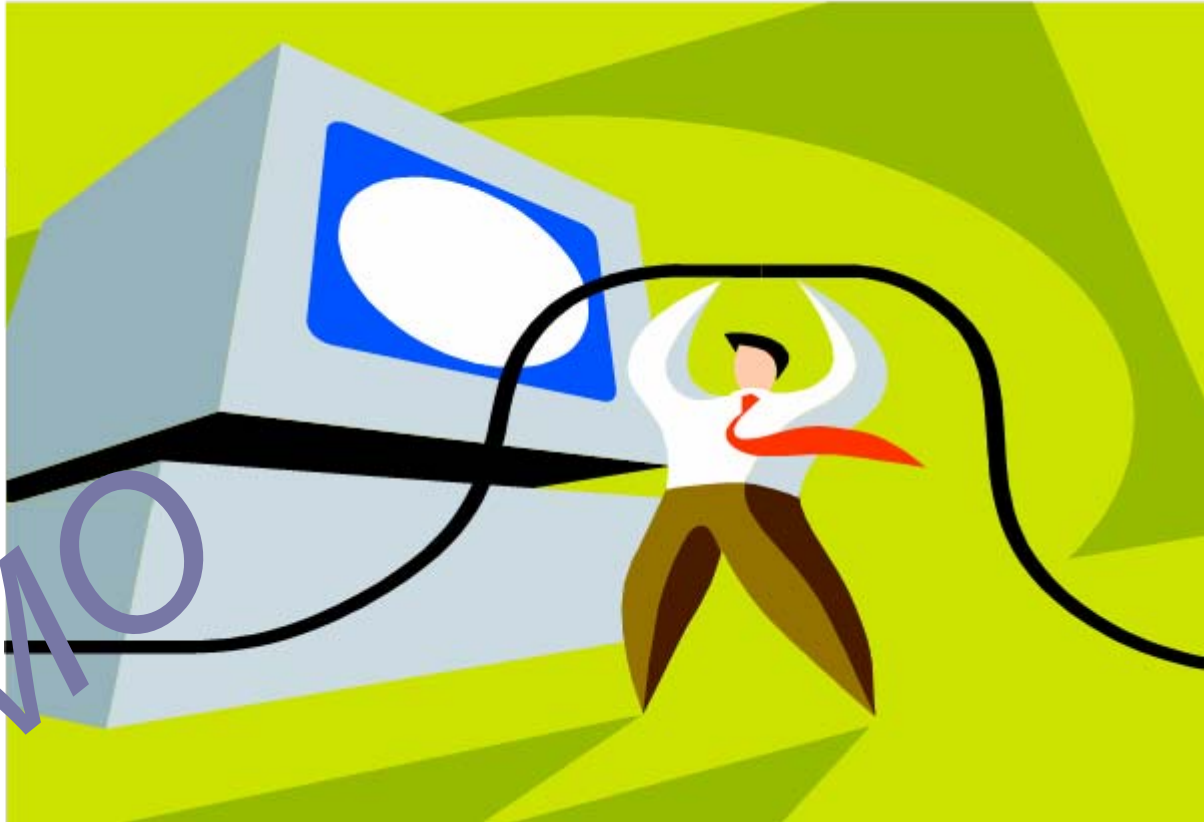
- Add option to right-click menu – “Configure” – in browsercustom.p, and launch special configuration window:

```
RUN SUPER.  
{get PopupActive lPopupActive}.  
IF lPopupActive THEN  
DO:  
    {get browseHandle hBrowse}  
    hPopupMenu = hBrowse:POPUP-MENU.  
  
    RUN createPopupItem IN TARGET-PROCEDURE (...  
  
    ON CHOOSE OF hBrowseConfig PERSISTENT RUN  
        configureBrowse IN TARGET-PROCEDURE.  
END.
```

Browse Configurator Method

- To enable “named” configurations that are not user-based, store a data string in another table
- Currently uses a “PLIP”, but these configurations should be cached in a manager (reduce appserver hits, improve performance)
- Profile manager stores user’s “named” configuration; if none, defaults to profile and then to design-time layout

DEMO



Deployment

- “all the activities that make a software system available for use” (Wikipedia)
 - Release
 - Install
 - Update
 - Uninstall
- Need to fill the gaps between Roundtable and Dynamics
- Worked with Thomas Hansen of appSolutions

Deployment Capability Comparison

<u>Deployment Requirement</u>	<u>Roundtable</u>	<u>Dynamics</u>
Release Control	Y	N
Static Objects	Y	N
Dynamic Objects (RYCSO)	Y	Y
Other Datasets (GSMMI, GSCER)	N	Y
Application Partitioning	N	Y
Packaging	Y	N
Database Versioning	Y	N
Installation Scripting	Y	N

Potential Solutions

- Enhance Roundtable Deployment (e.g. _update.w) to perform ado loads
- Use the DCU to apply a Roundtable release.

DCU Deployment

- Dynamics Configuration Utility needs to be configured to handle applications properly
- Main Points
 - Application database versioning
 - Management and control of delta's
 - DCU XML control files
 - ADO list generation
 - Putting it all together

Application Database Versioning

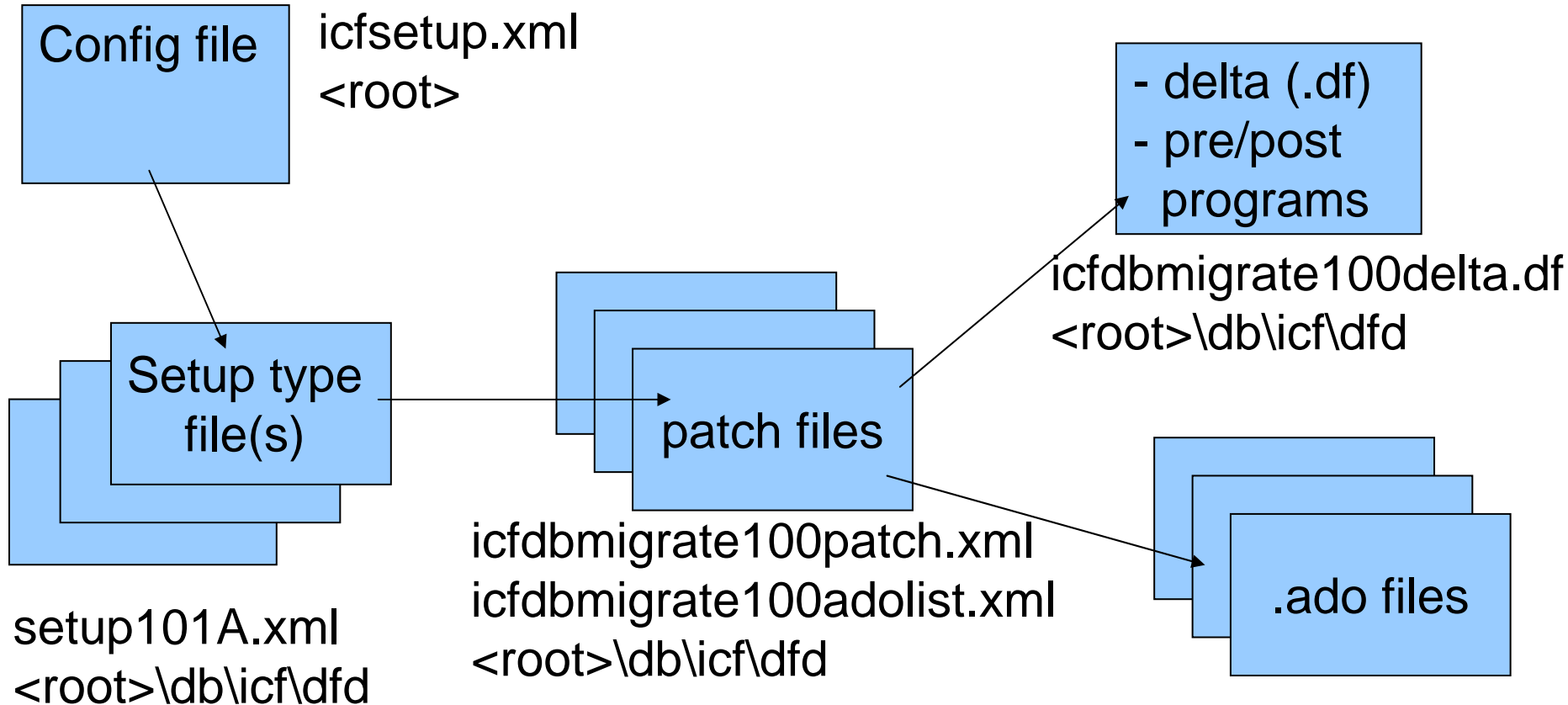
- Database version sequence – controls the versioning of the entire application
- *Seq_dbname_DBVersion*
- Applications with multiple databases should have a 'main' controlling database
- DCU upgrade must start from a known minimum level version.
- Controlled by *setup.XML

Example DCU Setup Shortcut

```
<...>\prowin32.exe -p icfcfg.w  
-icfparam  
DCUSETUPTYPE=Migrate21Setup  
(ICFCONFIG=icfsetup.xml is implied)
```

```
-icfparam  
ICFCONFIG=dtmssetup21b.xml,  
DCUSETUPTYPE=DTMSSetup
```

DCU Files



Overview of Setup Type XML file

1. Wizard Page Control

- Each page collects data from the human installer - ideally you just press “Next”
- Collects database connection parameters

2. Database Nodes

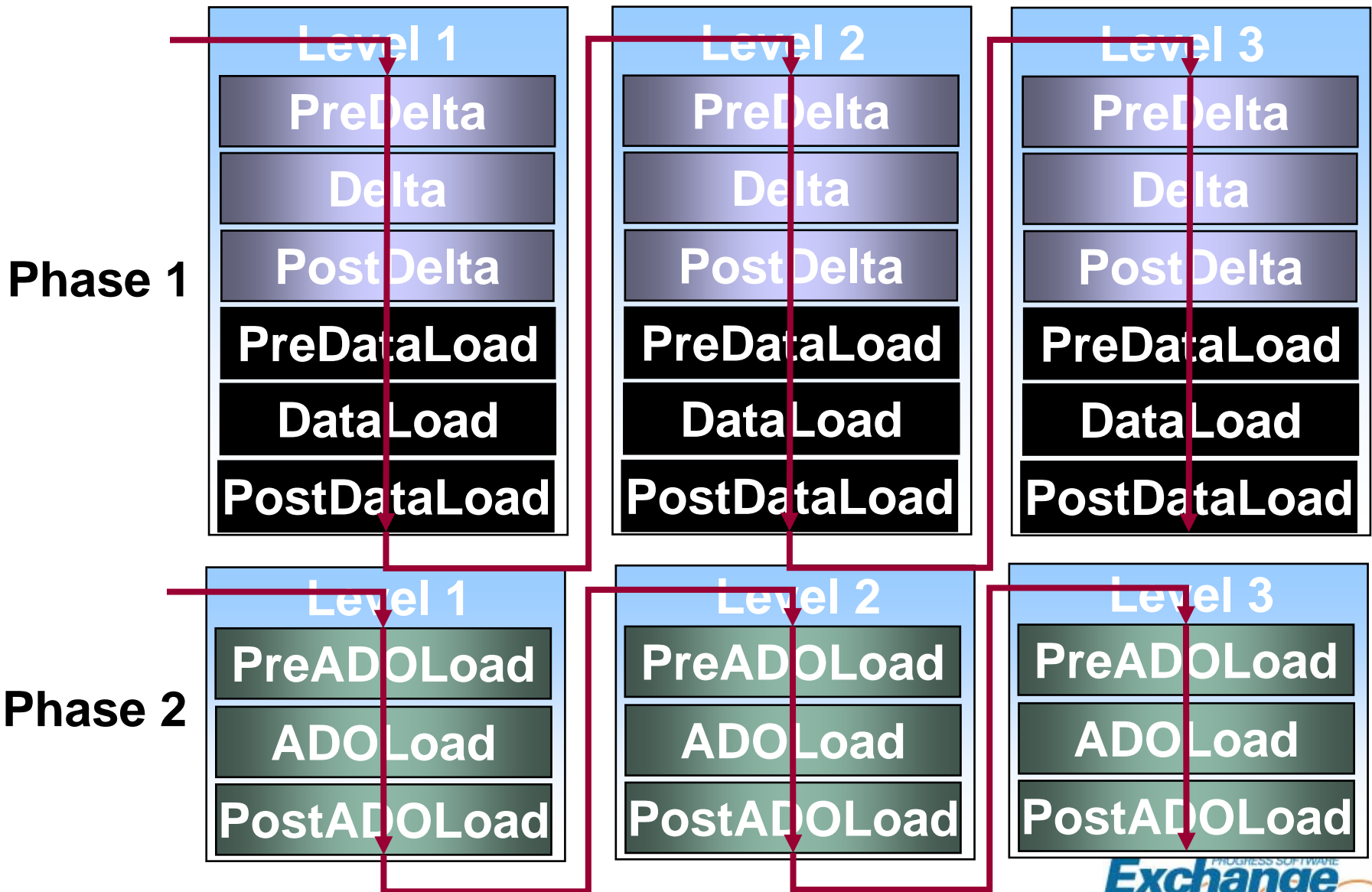
- Lists patch files to use
- Based on current database version

Setup Type XML File – Database Node

```
<patch PatchLevel="020029" NodeURL="db/icf/dfd/icfdb020029patch.xml" />
<patch PatchLevel="020029" NodeURL="db/icf/dfd/icfdb020029adolist.xml" />
</database>
- <database>
  <DBName>ACCT</DBName>
  <VersionSeq>seq_ACCT_DBVersion</VersionSeq>
  <MinimumVersion>020000</MinimumVersion>
  <ConnectParams>-1</ConnectParams>
  <DBDir>#path_db#\acctdb\acct.db</DBDir>
  <DBDump>#path_install#\db\acct\dump</DBDump>
  <patch PatchLevel="0" DBBuild="Yes" NodeURL="..\..\db\acct\dfd\acctbuild.xml" />
  <patch PatchLevel="020300" NodeURL="..\..\db\acct\dfd\acct020300adolist.xml" />
  <patch PatchLevel="020300" NodeURL="..\..\db\acct\dfd\acct020300patch.xml" />
```

```
<?xml version="1.0" encoding="utf-8" ?>
- <SetupInclude>
- <Patch PatchLevel="020301">
- <PatchStage Stage="Delta">
- <Program>
  <FileType>df</FileType>
  <FileName>../../db/acct/dfd/acct020301delta.df</FileName>
  <Description>Applying ACCT 020301Delta</Description>
  <Rerunnable>no</Rerunnable>
  <NewDB>no</NewDB>
  <ExistingDB>yes</ExistingDB>
  <UpdateMandatory>yes</UpdateMandatory>
</Program>
</PatchStage>
- <PatchStage Stage="PostDelta">
- <Program>
  <FileType>p</FileType>
  <FileName>../../db/acct/dfd/glchartkwinit.p</FileName>
  <Description>Update glchart Key Word field</Description>
  <Rerunnable>yes</Rerunnable>
  <NewDB>no</NewDB>
  <ExistingDB>yes</ExistingDB>
  <UpdateMandatory>no</UpdateMandatory>
```

DCU Processing



Helpful Tools (from appSolutions)

- Adolist generator: prepares an adolist.xml file based on the delta between 2 Roundtable releases
- Application partitioning tool: provides granular control (by code subtype) of client and server side code
- DCU detection of connected db's to limit user intervention

Steps to create a deployment

- Create “release objects”: delta’s, patch.xml files, pre/post processing utilities
- Create the final release and a deployment
- Generate the adolist file
- Stage the deployment
 - Copy into staging area
 - Run the DCU to update db’s
 - Run _update.w to compile the app

Applying a release to customer site

- Copy in the new release in full
- Run the DCU

Benefits-

- Mirrors a dynamics upgrade
- Can include application and dynamics version upgrade together
- r-code only – no compile required

Demo – Using the DCU to update an application



Summary of Methods

- Use the ADM2's *custom mechanism to create custom superprocedures
- Use the object class hierarchy to subtype new classes
- Create your own managers to provide standard application services and cache static data
- Implement your own profile types and/or keys to retain user preferences on all your screens

In Summary

- There are 'gaps'
- The framework can be extended – (its just code)
- There are 'best practices' to do this
- Ideally these extensions should be shared among us
- Value for Dynamics, ADM2, and straight 4GL users

Some Common Myths

- Myth: Dynamics and OpenEdge RA have nothing in common
- Fact: Dynamics is a partial implementation of OpenEdge RA, and much of OpenEdge RA has evolved from the Dynamics experience.
- Myth: Dynamics is rigid and inflexible
- Fact: It can and should be extended for your application's requirements
- Myth: You can't use Dynamics with OpenEdge Architect
- Fact: If you can use the 4GL, you can use Dynamics. Its just code.

Questions?



Tom Greenwood
Degama Systems Inc.
President, GTA-PUG
tomg@degama.com



DEV-4: Extending Progress Dynamics®



Thank You

- Thomas Hansen
- Wouter Dupre
- Greg Higgins and the PEG
- The Dynamics Development team
- Gord Campbell
- Many others...